

Deformation Analysis  
of  
Rotary Combustion Engine Housings

Final Report  
on  
NASA Grant NAG 3-456

by  
Carl Vilmann  
Michigan Technological University

(NASA-CR-188187) DEFORMATION ANALYSIS OF  
ROTARY COMBUSTION ENGINE HOUSINGS Final  
Report (Michigan Technological Univ.)  
361 p

N91-27158

CSCL 21E

Unclas  
G3/07 0012362

NASA Grant NAG 3-456, "Deformation Analysis of Rotary Combustion Engine Housings," was initiated with three objectives in mind. The first of these objectives was the generation of a detailed finite element model of a rotary engine's center (trochoid) housing. Once this model had been generated, the second objective of the project, the prediction of the stress and deformation fields within the trochoid housing during engine operation, could be attacked. Finally, the third objective of this work was the development of a preprocessor which would simplify the generation of subsequent finite element models for alternate center housing designs. The purpose of this preprocessor was to greatly shorten the development time for modified trochoid housings. While the first and third objectives were fully met, the second objective proved difficult to achieve. The following report details the work done at MTU on NASA Grant NAG 3-456.

## Objective 1

### **Development and Verification of a Finite Element Model the Trochoid Housing**

Initially the engine type under investigation was manufactured by Mazda. A picture of this engine's trochoid housing is included as Figure 1. The final finite element model for this housing is

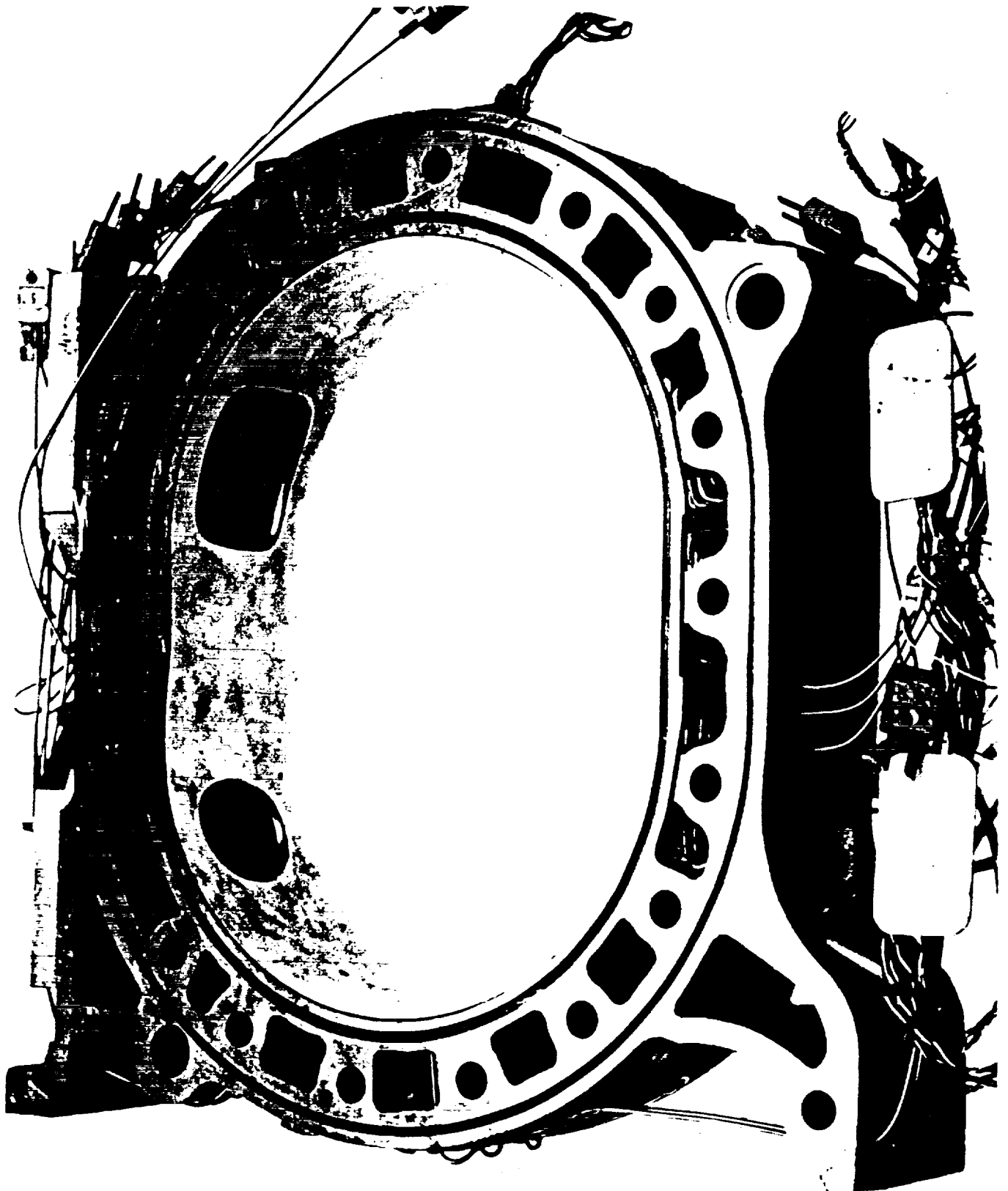


Figure 1

Mazda Type Trochoid Housing

shown in Figure 2. This model was developed and tested as the M.S. thesis of Scott Bradley. During Scott's development process various element types were examined for their suitability and different mesh densities were explored to determine the coarsest mesh which would model the housing's response accurately. In order to verify the response of the finite element model, a comparison was made with experimentally measured stresses reduced from strain gage measurements. The discrepancy between the experimental results and the finite element predictions were found to be acceptable, and the model shown in Figure 2 was accepted for use in the dynamic simulation phase of the project. A more detailed description of the development and testing of this model is contained in Scott's M.S. thesis which is attached as Appendix A.

## Objective 2

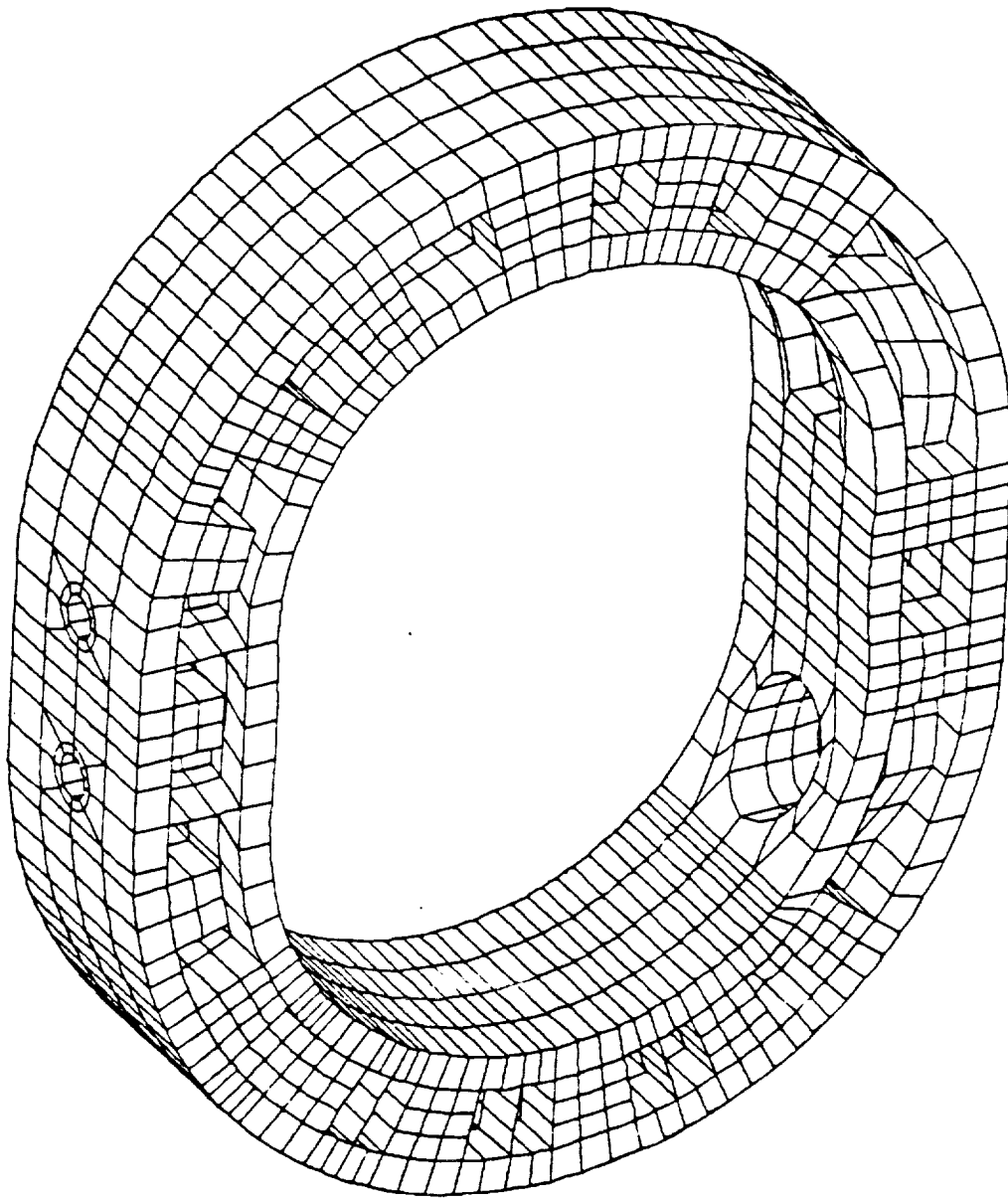
### **Prediction of the Stress and Deformation Fields Present within the Trochoid Housing During Operating Conditions**

At this point in the project NASA changed the engine under investigation from the Mazda engine to an engine produced by Outboard Marine Corporation (The OMC engine). The geometric similarities between the Mazda engine and the OMC engine were few. While the basic trochoidal shape of the bore remained the same, the cooling channel pattern was radically different. This change



required the complete regeneration of a finite element model. The information on mesh refinement gained from the development of a FEM model for the Mazda engine was useful, but the input of the geometrical characteristics needed to be completely redone. An M.S. student was enlisted to generate and exercise a model of the OMC engine. Unfortunately, this student left after one year without making much progress. A second student (A M.S. student who remained at MTU after finishing his degree.) with FEM experience was employed to help with the modelling. A finite element model was generated and delivered after nine months. When an attempt was made to exercise this model, it was discovered that elements with unacceptable geometric shapes were present within the model. In order to eliminate these unacceptable elements, the project director revised the model. The final finite element model of the OMC center housing is shown in Figure 3. In viewing this model it should be noted that since the trochoid housing is symmetric about its midplane, only 1/2 of the housing needed to be modeled.

With the model complete, the next step in the prediction of the stress and deformation fields present during operating conditions was the calculation of the model shapes and natural frequencies of the housing. The boundary conditions necessary were generated, and the job was submitted to the MTU computational facility. Unfortunately, the operating system present on the MTU computer did



**Figure 2**  
**Finite Element Mesh**  
**for the**  
**Mazda Type Trochoid Housing**

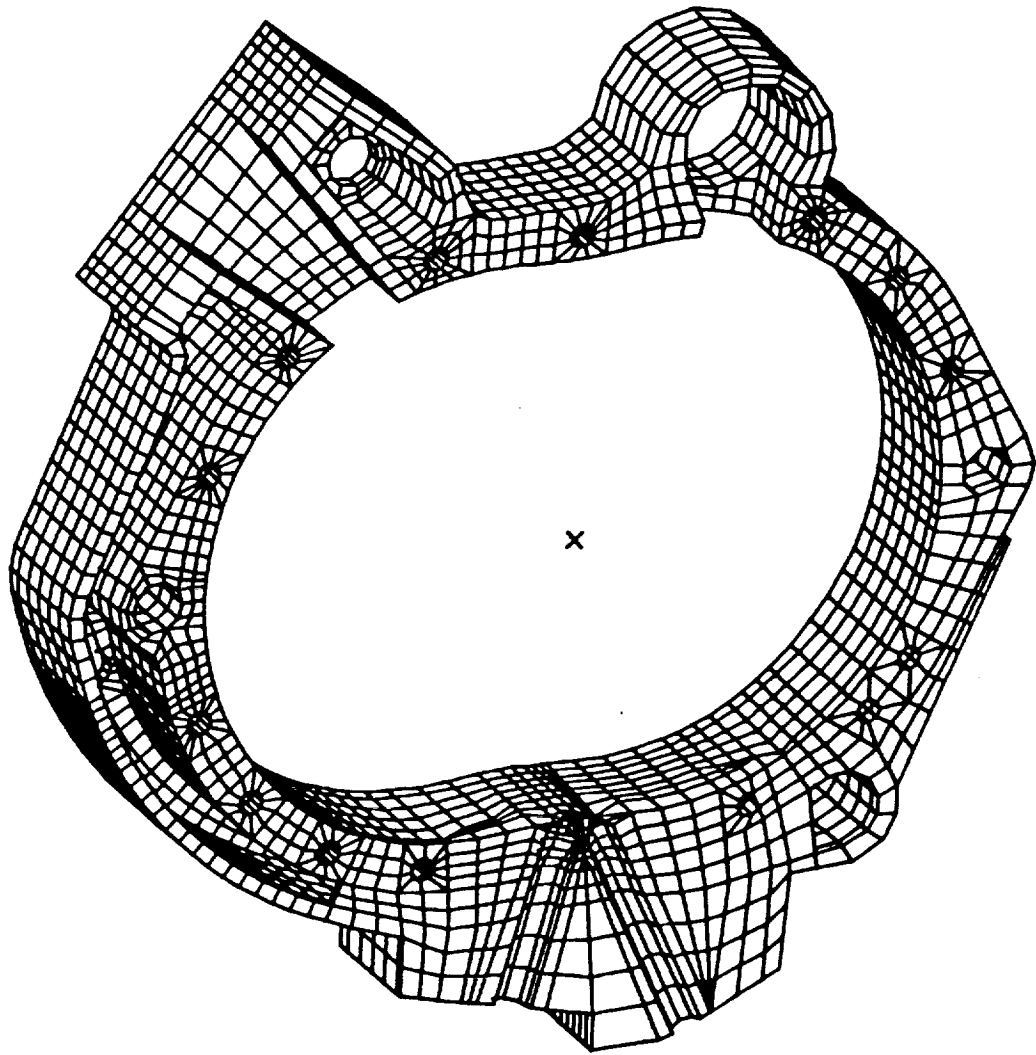


Figure 3  
Finite Element Mesh  
for the  
OMC Type Trochoid Housing

not allow file lengths of sufficient magnitude to allow calculation of the dynamic characteristics of this model. After conferring with NASTRAN consultants, a fix to this problem was discovered. The model needed to be divided into substructures. These subsets of the original model would then be treated separately and finally computationally recombined, thereby eliminating the file length problem. This substructuring was performed by the project director, and the revised model was once again submitted. Time estimates for completion of the job calculated by NASTRAN were on the order of 150 hours. Since it is virtually impossible to complete a job of that length without the MTU system going down, it was decided that competing this work with the computational facility at MTU was unfeasible.

At this point in the project a NASA Cray account was requested and received. The FEM model of the OMC housing was loaded on tape and forwarded to Lewis Research Center. The eigenvalue problem associated with determining the mode shapes and natural frequencies was completed and the lowest natural frequencies calculated are listed are shown in Table 1.

The modal results were then ready for recombination in order to determine the stress history at the placed where the experimental data had been extracted. This required the transfer of the eigenvalue results back to MTU. At this point another obstacle was encountered. The file transfer routines required for the transfer

**Table 1**  
**Lowest Natural Frequencies**  
**for the**  
**Trochoid Housing**

Mode	Natural Frequency (Cycles/Sec)
1	1621
2	1908
3	1981
4	2458

of large amounts of data between MTU and NASA were not working correctly. The project director spent approximately one month trying to find a solution to this problem and finally obtained a band-aid, temporary, fix to the data transfer problem.

This is where the dynamic analysis is at the present time. The delays have been unfortunate. The work remaining on the dynamic analysis includes the recombination of the modal response to the pressure loading on the housing, and the superposition of these results with the thermal stresses arising from the temperature distribution present within the engine during operating conditions. Since the final report is now being requested, these tasks remain unfinished.

### Objective 3

#### **Development of a Specialized Preprocessor for the Trochoid Housing**

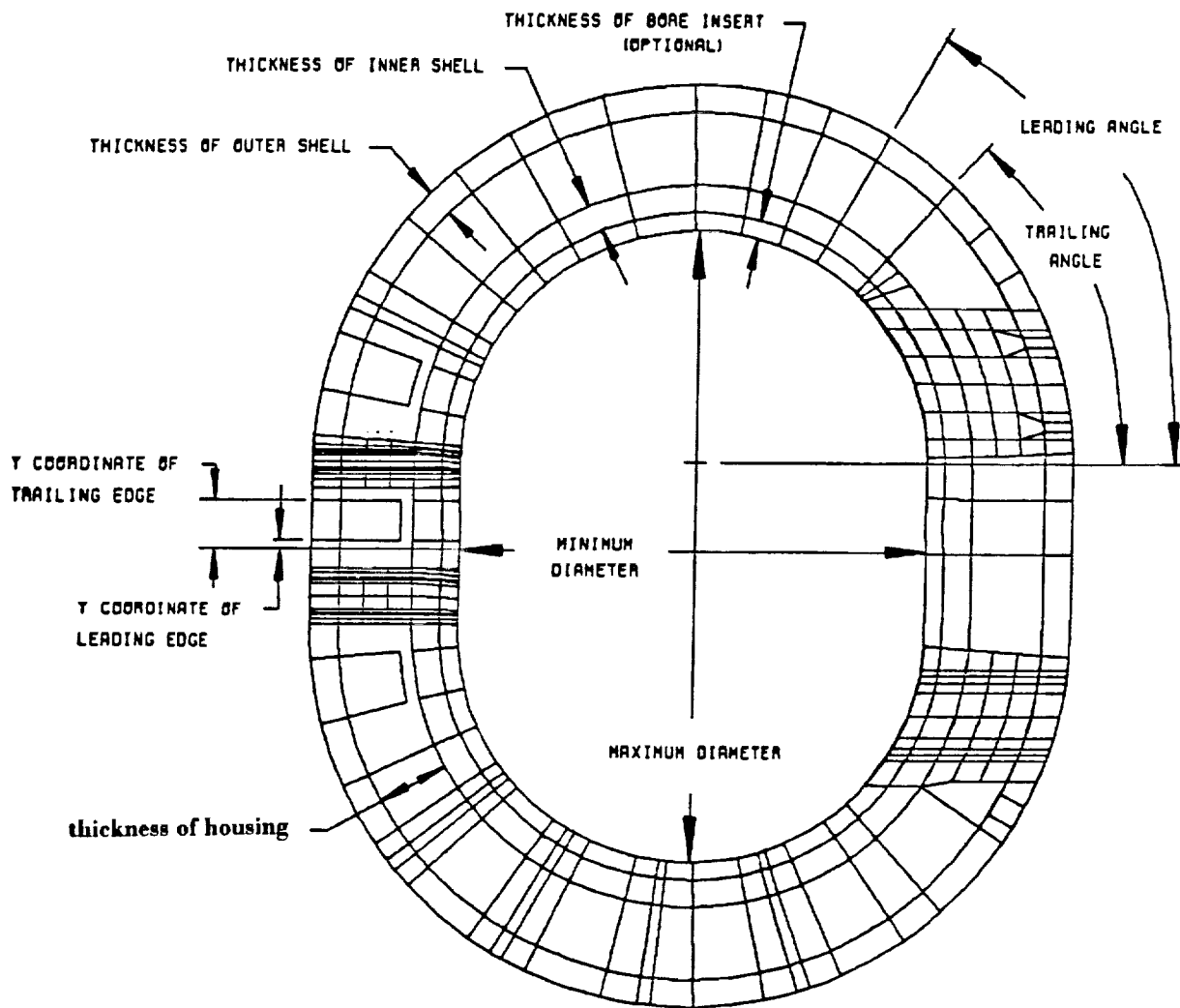
The objective of this portion of the project was to develop a software product which would simplify the preparation of finite element models for a rotary engine's trochoid housing. Since a Mazda engine was initially under consideration, and since there were no indications early in the project that a switch to an alternate engine type would be desired or necessary, the preprocessor developed generates the complete finite element mesh for a rotary engine center housing of the Mazda type. This preprocessor shortens the development time necessary for mesh generation of a trochoid housing's FEM model from roughly one man month to approximately two man hours. The creation of the is preprocessor was the M.S. thesis of W. Lychuk.

In developing this preprocessor it was decided that duplication of the commercially available software for generating a finite element mesh within a defined geometric volume was unnecessary. Therefore, the preprocessor developed was written to prepare data for the commercial preprocessor SUPERTAB. SUPERTAB, now part of the IDEAS package available from Structural Dynamics Research Corporation was in place both at NASA and at Michigan

Tech. The Michigan Tech preprocessor generates a program file for SUPERTAB. After the input parameters detailed in Figure 4 are supplied to the MTU preprocessing software, an output file results which can be used to drive SUPERTAB to complete a finite element model of the housing. The housing parameters which can be altered include:

- 1) The maximum and minimum diameters of the trochoidal bore.
- 2) The thickness and axial depth of the center housing.
- 3) The thickness of both the inner and outer shells.
- 4) The thickness of the bore insert.
- 5) The size, type, and location of each rib.
- 6) The type of each channel.
- 7) The size and location of the intake port.
- 8) The size and location of the exhaust port.
- 9) The size and location of the spark plug ports.

The details of the development of this preprocessor are contained in the M.S. thesis of W.M. Lychuk. A copy of this thesis is attached as Appendix B. A listing of the code developed as the aforementioned preprocessor is included as Appendix C.



**Figure 4**

**Input Parameters  
for the  
MTU Preprocessor**



## Conclusion

Problems with personal and computational facilities were responsible for this project falling short of all of its originally stated objectives. In spite of these problems, detailed, executable, finite element models were developed for both the Mazda and the OMC trochoid housings. In addition, it was demonstrated that a preprocessor which would hasten the generation of finite element models of a rotary engine could be developed. Two publications resulted from this grant and summarize the work detailed in Appendices A & B. These publications are:

- 1) "Finite Element Model Development of Multiple Rotary Combustion Engine Housing Configurations," by C.E. Passerello, C.R.Vilman, S.A. Bradley, and W.M. Lychuk, published in the Proceedings of the CADAM User Exchange, May 1985 meeting in New Orleans, LA.
- 2) "Stress and Deformation Modeling of Multiple Rotary Combustion Engine Trochoid Housings," by W.M. Lychuk, S.A. Bradley, C.R. Vilman, C.E. Passerello, and C. Lee, SAE paper number 860614, published in the

proceedings of the SAE International Congress and  
Exposition, Feb. 1986 meeting in Detroit MI.

While all of the initial objectives were not met, they were, possibly, too ambitious given the constraints that were present. As a final note, the project director would like to express his disappointment in not being able to completely fulfill the original goals, and thank NASA for their support of this project.

# **Appendix**

**A**

FINITE ELEMENT MODELING OF A WANKEL  
ENGINE CENTER HOUSING

by  
Scott Bradley

A THESIS  
Submitted in partial fulfillment of the requirements  
for the degree of  
MASTER OF SCIENCE IN ENGINEERING MECHANICS

MICHIGAN TECHNOLOGICAL UNIVERSITY

1985

**LIBRARY**  
MICHIGAN TECHNOLOGICAL UNIVERSITY  
HOUGHTON, MICHIGAN

## ABSTRACT

In this study, construction of a finite element model of a Wankel rotary combustion engine's center housing is presented. Analysis of the model is purely static, however, thermal and dynamic analysis considerations are taken into account. Verification of the model is accomplished through experimental tests on the actual housing. A convergence study is also completed by varying element types and interpolation orders. From the analysis and subsequent testing, it is believed that construction of a valid FEM of a Wankel engine's center housing has been completed.

## ACKNOWLEDGEMENTS

The author would like to thank his advisor, Dr. Chris E. Passerello, for his invaluable assistance in preparing this thesis.

Thanks is also expressed to Dr. Carl R. Vilmann, Dr. L. Bouge Sandberg, and Dr. Madhukar Vable for taking time out from their busy schedules to review and examine the paper.

Further graditude is extended to the employees in the Michigan Technological University's graphics lab. Their expertise provided the necessary software to execute this study.

In addition, the author sincerely extends thanks to his parents. Without their confidence and support this study could not have been completed.

Most of all, to his wife, who never lost sight of their goals, the author expresses his undying love.

## TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	v
1. BACKGROUND	1
1.1 Basic Operation	1
1.2 Comparison to a Piston Engine	4
1.3 Problems Plaguing the Rotary Engine	4
1.4 Center Housing Geometry	5
1.5 Tricohoidal Bore	7
2. MODEL EXPECTATIONS	11
2.1 Modeling Requirements	12
2.2 Modeling Capabilities	12
2.3 Model Construction	15
3. CENTER HOUSE TESTING	18
3.1 Uniform Internal Pressure Test (IPT)	18
3.2 Discussion of the Tensile Test	23
3.3 Simulation of the Tensile Test	26
4. CONSTRUCTION METHODS AND MODEL VALIDITY	37
4.1 Construction of the Linear Thin shell Model	37
4.11 Analysis of the Linear Thin Shell Model	42
4.2 Construction of the Parabolic Thin Shell model	44
4.21 Analysis of the Parabolic Thin Shell Model	58
4.3 Construction of the Parabolic Thick Shell Model	64
4.31 Analysis of the Parabolic Thick Shell Model	67
5. PRESENTATION AND REVIEW OF THE FINAL MODEL	73
5.1 Construction	73
5.2 Performance and Ability	77

6. CONCLUSIONS AND RECOMMENDATIONS	83
APPENDIX 1. SUPERB FINITE ELEMENT LIBRARY	
A1.0 Stress Output (Coordinate System Orientation)	84
A1.1 Thin Shell Element	86
A1.2 Thick Shell Element	89
A1.3 Solid Element	92
A1.4 Beam Element	92
APPENDIX 2. ELEMENT BENCHMARK TESTS	
A2.0 Shell Element Tests	94
A2.1 Solid Element Tests	97
APPENDIX 3. EXACT SOLUTION FOR BENCHMARK TESTS	
APPENDIX 4. NUMERICAL ROUTINES USED IN FEM CONSTRUCTION	
REFERENCES	103



# LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.1	View of rotor and center housing	2
1.2	Full engine configuration	2
1.3	Combustion cycle of Wankel engine	3
1.4	Actual housing supplied by NASA	6
1.5	Coolant flow diagram	7
1.6	Illustration of the various regions of the housing	8
1.7	Illustration of epitrochoid generation	9
1.8	Illustration of trochoid generation	9
2.1	SUPERTAB generation levels	14
3.1	Illustration of the internal pressure test and locations of the strain gage rosettes	19
3.2	Internal pressure test (location n)	20
3.3	Internal pressure test (location h)	21
3.4	Internal pressure test (location x)	22
3.5	Simulated endplate deformation	24
3.6	Illustration of turnbuckles used to test housing	25
3.7	Longitudinal tensile test	25
3.8	Transverse tensile test	25
3.9	Longitudinal tensile test (location n)	27
3.10	Longitudinal tensile test (location h)	28
3.11	Longitudinal tensile test (location x)	29
3.12	Transverse tensile test (location n)	30
3.13	Transverse tensile test (location h)	31
3.14	Transverse tensile test (location x)	32
3.15	Illustration of suspected bolt deformation	33
3.16	Location of model support	35
3.17(a)	Thin shell model's support scheme	36
3.17(b)	Thick shell model's support scheme	36
4.1	Illustration of linear thin shell model	38
4.2	Illustration of coincident node requirement	40
4.3	Illustration of mid-plane stiffener location	40
4.4	Illustration of possible deformations by utilizing the coupling method	42
4.5	Various regions where support of the model was investigated	43
4.6	Deformation of linear thin shell model (support case 1)	45
4.7	Deformation of linear thin shell model (support case 2)	46
4.8	Deformation of linear thin shell model (support case 3)	47
4.9	Stresses at location n (linear thin shell model) (restraint case 1)	48
4.10	Stresses at location n (linear thin shell model) (restraint case 2)	49
4.11	Stresses at location n (linear thin shell model) (restraint case 3)	50
4.12	Stresses at location h (linear thin shell model) (restraint case 1)	51

4.13	Stresses at location h (linear thin shell model) (restraint case 2)	52
4.14	Stresses at location h (linear thin shell model) (restraint case 3)	53
4.15	Stresses at location x (linear thin shell model) (restraint case 1)	54
4.16	Stresses at location x (linear thin shell model) (restraint case 2)	55
4.17	Stresses at location x (linear thin shell model) (restraint case 3)	56
4.18	Illustration of strain gage rosette and stress grid locations	57
4.19	Deformation of the parabolic thin shell model subjected to a 1000 lb. tensile load	60
4.20	Stresses at location n (parabolic thin shell model) (restraint case 2)	61
4.21	Stresses at location h (parabolic thin shell model) (restraint case 2)	62
4.22	Stresses at location x (parabolic thin shell model) (restraint case 2)	63
4.23	Illustration of the parabolic thick shell model	65
4.24	Illustration of possible deformations of housing without elemental rotations supported	66
4.25	Illustration of the introduction of beam elements	66
4.26	Stresses at location n (parabolic thick shell model) (restraint case 2)	68
4.27	Stresses at location h (parabolic thick shell model) (restraint case 2)	69
4.28	Stresses at location x (parabolic thick shell model) (restraint case 3)	70
4.29	Deformation of the parabolic thick shell model subjected to a load of 1000 lbs.	71
5.1	Illustration of the final FEM of the housing	74
5.2	Illustration of exhaust port mesh	75
5.3	Illustration of intake port mesh	75
5.4(a)	Illustration of spark plug region (view of the inner bore)	76
5.4(b)	Illustration of spark plug region (view of the outer surface)	76
5.5	Stresses at location n (final FEM of the housing) (restraint case 2)	78
5.6	Stresses at location h (final FEM of the housing) (restraint case 2)	79
5.7	Stresses at location x (final FEM of the housing) (restraint case 2)	80
5.8	Deformation of final FEM subjected to a 1000 lb. tensile load	81
A1.1	Typical thin shell element stress output	85
A1.2	Typical thick shell element stress output	85
A1.3	Typical solid element stress output	86
A1.4	Various thin shell elements available	87
A1.5	Thin shell surface definition	87
A1.6	Local axis orientation	88
A1.7	Elemental axis orientation	89
A1.8	Various thick shell elements available	90
A1.9	Thick shell element surface definition	91
A1.10	Various solid elements available	92
A2.1	Illustration of simply-supported plate test	95

A2.2	Illustration of clamped plate test	95
A2.3	Illustration of beam test	97
A2.4	Illustration of column test	98
A4.1	Illustration of the construction point placement	102

## 1.00 BACKGROUND

The Wankel rotary combustion engine (RCE) was initially introduced to the automotive industry in the early 1960's. Praised as the engine of the future, it is basically an internal combustion engine which operates on the otto cycle: intake, compression, combustion, and exhaust. It executes the cycle by utilizing a triangular shaped rotor within a trochoidal shaped working chamber, both of which are shown in figure 1.1. These working chambers, commonly known as the center housings, are located along the mainshaft and are separated by endcovers. The full engine configuration is shown in figure 1.2. Utilized in a variety of environments, the Wankel is a lightweight, compact, smooth running engine which has multifuel capability and can be stratified charged very easily (1).

### 1.10 BASIC OPERATION

In the RCE, each face of the rotor can be considered a combustion chamber sliding along the inner bore. By rotating the rotor within the bore, the otto cycle can be simultaneously completed three times for every revolution of the rotor. This process is illustrated in figure 1.3. The combustion phase exerts pressure on one face of the rotor which produces a rotary motion and brings the next chamber into firing position. In order to produce the rotary motion, the pressure from combustion must be directed on a point away from the mainshaft centerline. For this reason, the rotor is mounted on eccentric bearings which provide the leverage required. Phasing between the rotor rotation and the eccentric bearing rotation is accomplished through the use of phasing gears. A part of this gearing is a

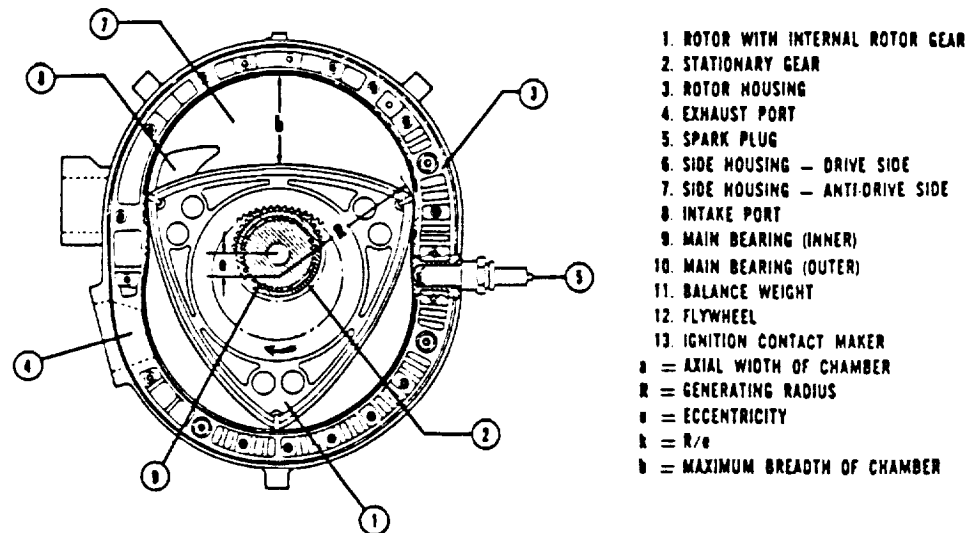


Figure 1.1: View of rotor and center housing.  
 (Courtesy of Chilton Book Co.)

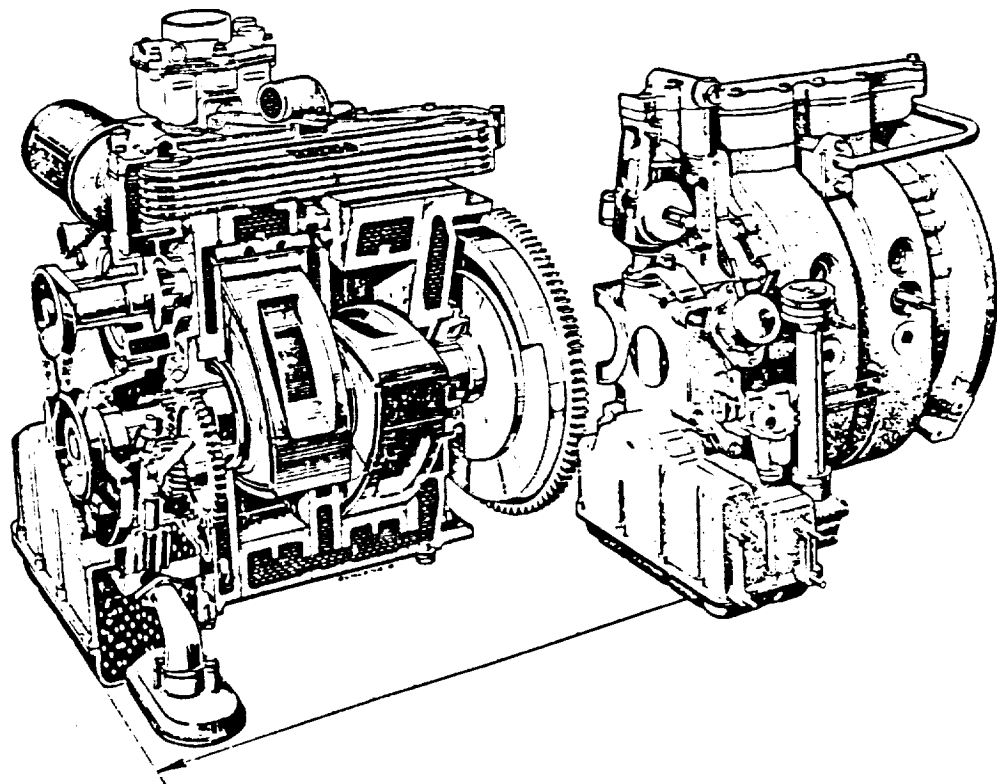


Figure 1.2: Full engine configuration.  
 (Courtesy of Chilton Book Co.)

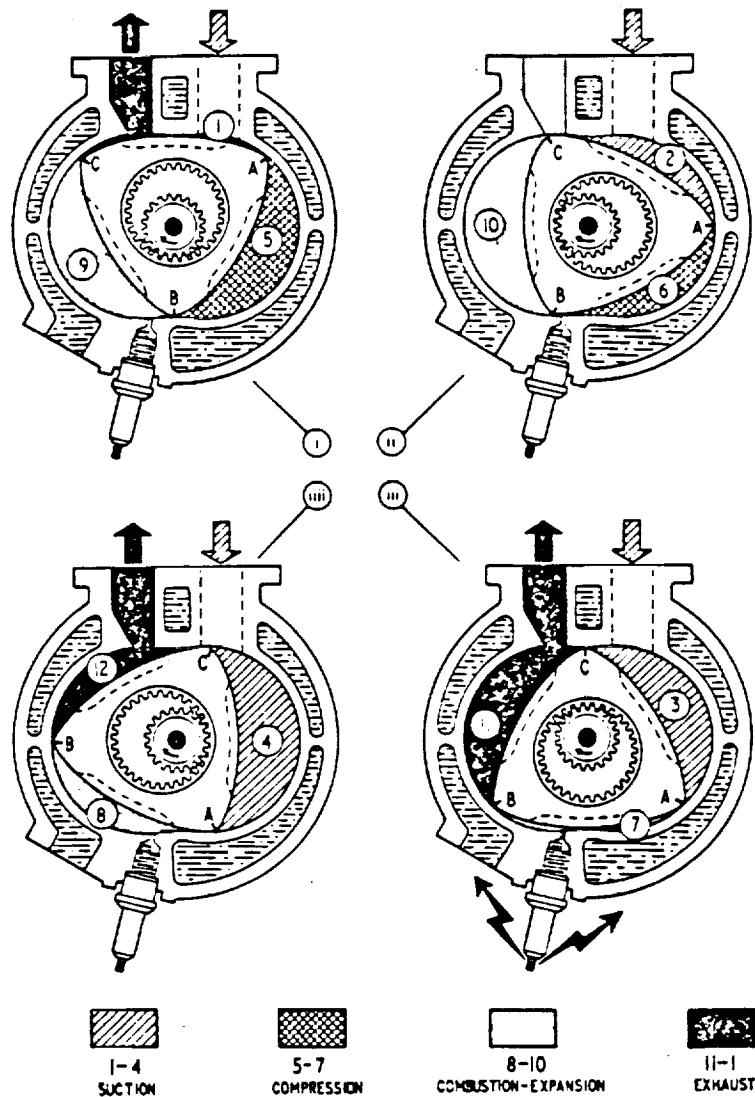


Figure 1.3: Combustion cycle of Wankel engine.

(Courtesy of A. S. Barnes & Co.)

stationary reaction gear mounted to the end cover and concentric to the mainshaft. This reaction gear meshes with an inner ring gear on the rotor. A gear ratio of 3:2 must exist. This gear ratio assures that for every rotation of the rotor, the mainshaft completes three rotations. Thus, a positive torque is applied for every one-third rotation of the rotor which gives one complete power phase for each rotation of the mainshaft.

## 1.20 COMPARISON TO A PISTON ENGINE

When comparing the rotary combustion engine (RCE) to a reciprocating engine (RPE), certain distinctions are clear. Instead of utilizing a cylindrical piston and its associated connecting rod, the RCE uses a rotor centered around a mainshaft. Also, ports, which are opened and closed by the motion of the rotor, replace the complicated valves and valve trains of the RPE. With the connecting rods, camshafts, pushrods, and the other assorted parts eliminated, the RCE's size and weight is reduced to about half that of a RPE with comparable power output. In addition to fewer parts, there are fewer moving parts, indicating that the RCE operates at a lower manufacturing and repair cost. Fewer moving parts also produce less wear and power loss so reliability is high.

Furthermore, the use of a rotor eliminates the large inertia forces created by reciprocating masses. This, in addition to the application of positive torque over a full mainshaft rotation, leads to a smoother operating engine. Gas pressure loads on the bearings are higher in the RCE than in a piston engine. However, centrifugal loads are lower, thus the risk of bearing failure is far smaller which again makes the rotary engine highly reliable. Overall, the rotary combustion engine is theoretically far superior to the reciprocating piston engine.

## 1.30 PROBLEMS PLAGUING THE ROTARY ENGINE

Although the rotary engine's performance should be superior to a piston engine, actual operation and development have been plagued with problems. A 12% higher specific fuel consumption, higher hydrocarbon emissions, and excessive wear of the center housing bore have hampered the success of the engine. The cause of these problems seem to be centered around the sealing system,

specifically the apex seal. The apex seals which are located on each tip of the rotor, tend to leave the inner trochoidal bore's surface just after ignition (2). When the seal retains contact with the surface it skids before complete semifrictionless contact is acquired. This liftoff and the subsequent skidding is thought to produce the effects discussed earlier. Many theories have been formulated on why apex seal liftoff occurs. Breakdown of the lubricating agents, vibration of the seal within its setting, and actual housing deformation are suspected causes. Changes in material and geometry can reduce effects but further study is required.

A few studies have been done on the effects of modifying the lubricating agents and seal properties: Behling and Weise (4) and Rogers, et. al. (5). There also have been studies performed on apex seal vibration: Prasse, McCormick, and Anderson (6), Matura, et. al. (2) (7). Yet no study has been found in which a finite element deformation analysis of the center housing was performed. The need for a user friendly finite element model of the housing, which can be modified to encompass geometry and loading changes is therefore needed. The scope of such a project, though, is too large to be completed by this thesis. Thus, it is the purpose of this thesis to lay down a foundation on which future study can be done. This thesis will present the construction and validation of a finite element model of the housing alone. The model will be verified using static loading conditions with experimentally collected data.

#### 1.40 CENTER HOUSING GEOMETRY

The housing used in this study, shown in figure 1.4, was provided by the NASA Lewis Research Center. It was obtained from a custom built, 573 cc (35 cubic inch), high performance single rotor engine. The housing is centered about the mainshaft and is sandwiched between two endplates. It is made of an aluminum alloy and has a steel liner with a thickness of approximately 1 mm within the trochoidal shaped bore. Cooling channels pass



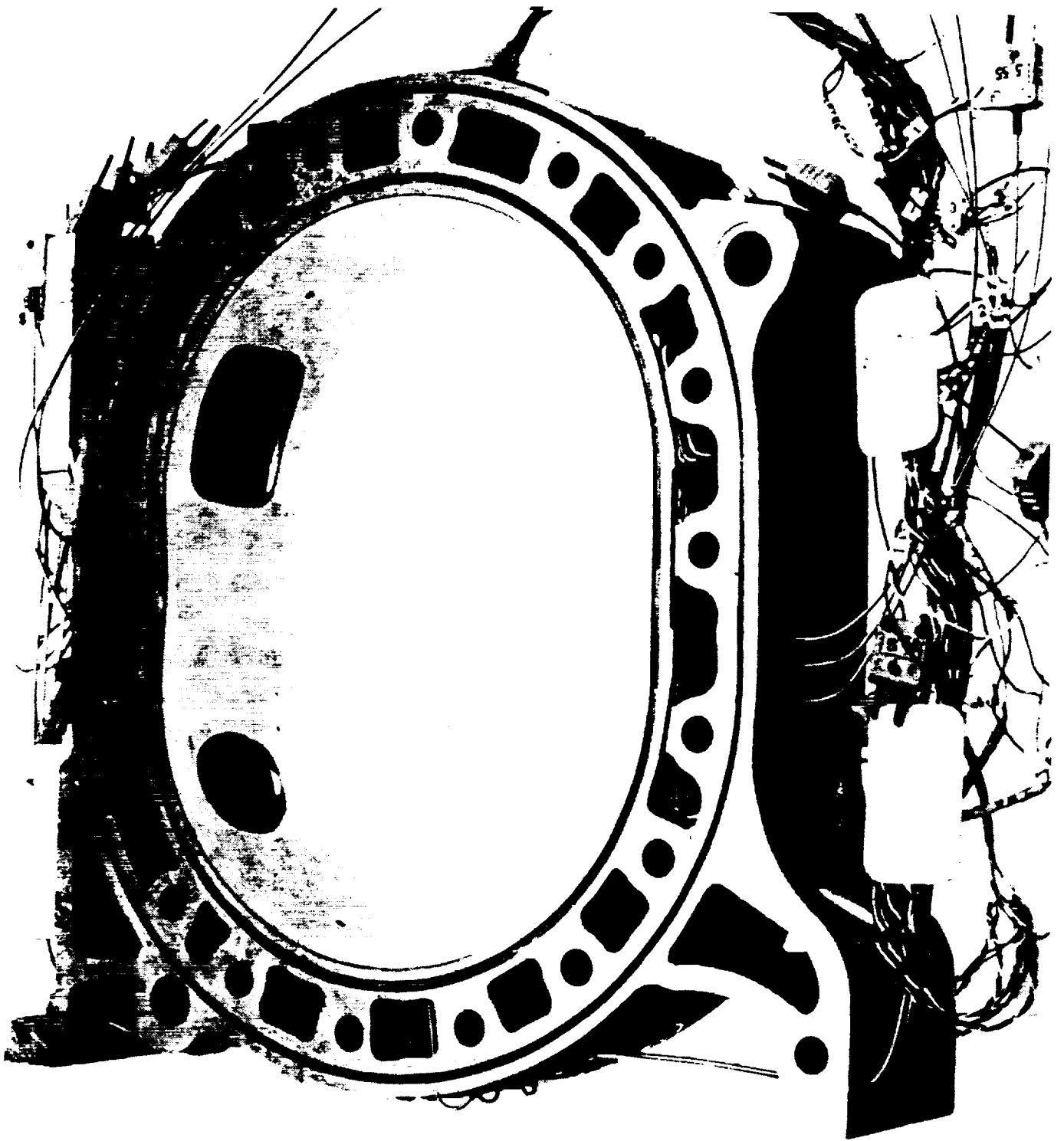


Figure 1.4: Actual housing supplied by NASA.

ORIGINAL PAGE IS  
OF POOR QUALITY

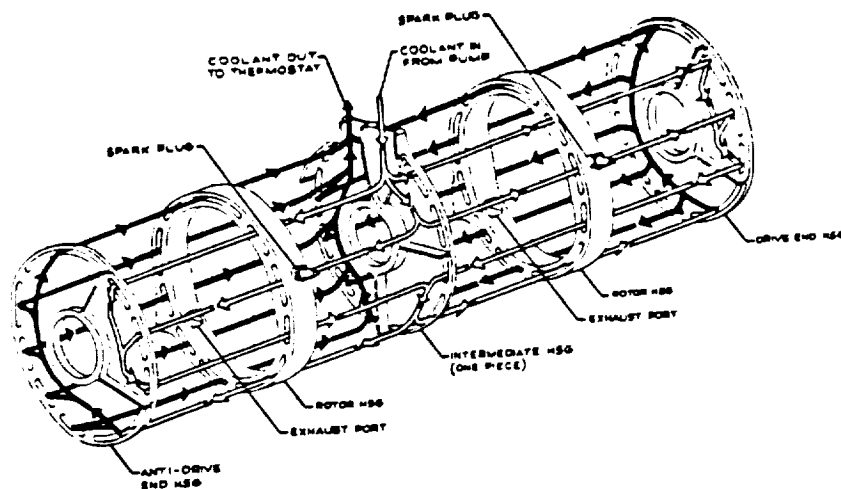


Figure 1.5: Coolant flow diagram.  
(Courtesy of Chilton Book Co.)

through the housing to allow axial flow of coolant as shown in figure 1.5. Noticeable features of the housing include the intake and exhaust ports, spark plug locations, bolt holes, and engine mounts. An added feature is the inner surface around the spark plug region. In order to aid in thermal diffusion, the surface is thinned. Terminology that will be used to describe the housing is illustrated in figure 1.6.

#### 1.50 TROCHODIAL BORE

A prominent feature of the Wankel RC2 is the shape of the inner bore. Although it is of trochoidal shape, the generation of the trochoid is first accomplished by constructing a true epitrochoid. Since the definition of an epitrochoid is the loci of a point on the radius of a circle which rolls without slip around a fixed base circle, see figure 1.7, an infinite number of epitrochoids are available. Only by making the base circle's radius twice that of the rolling circle's radius, is the familiar two lobed epitrochoid obtained. Thus, the basic dimensions needed to describe the epitrochoid are the base circle radius ( $R_1$ ), the rolling circle radius ( $R_2 = R_1/2$ ), and the distance of the generating point from the center of the rolling circle, or the eccentricity ( $e$ ).

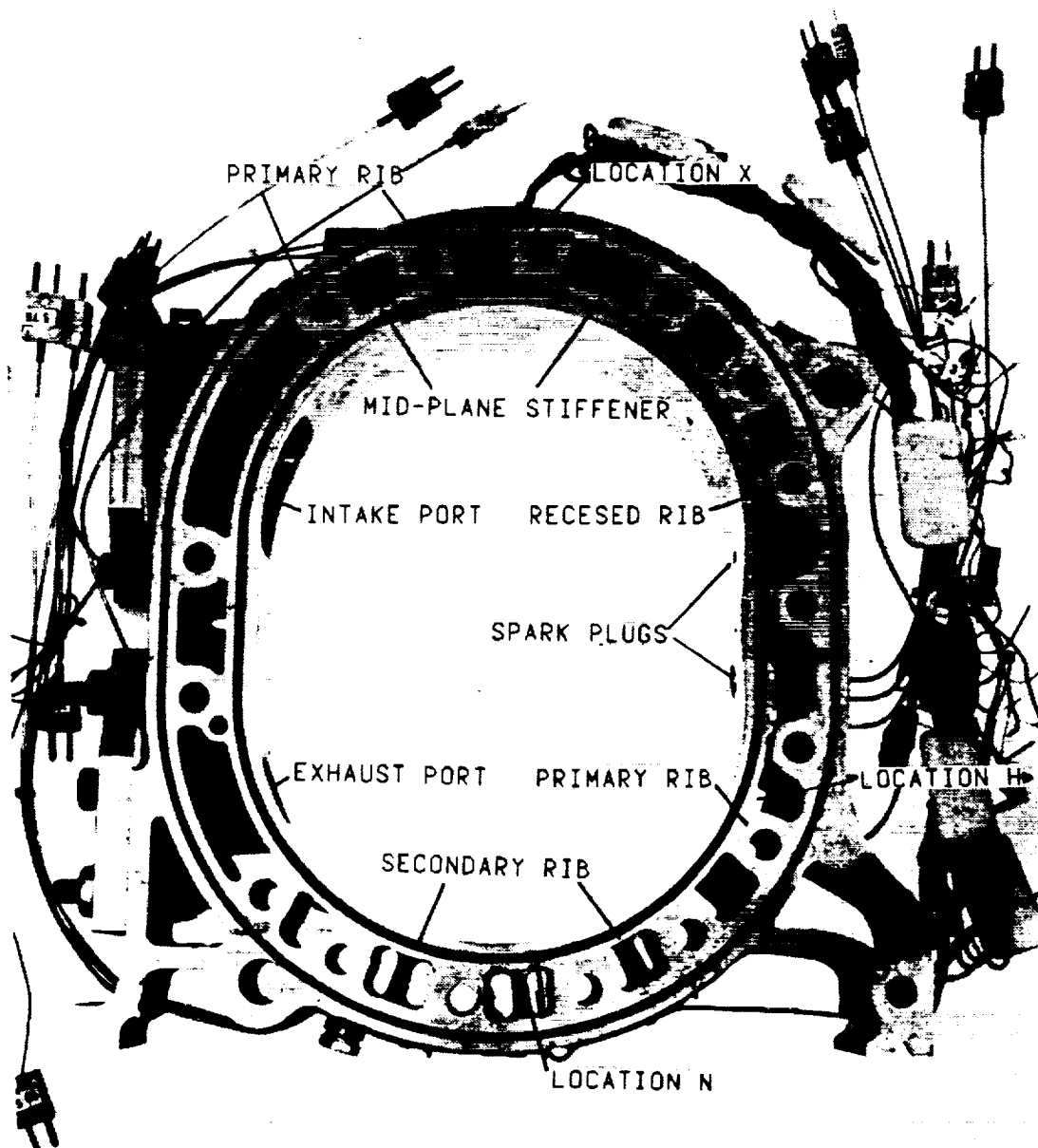


Figure 1.6: Illustration of the various regions of the housing.

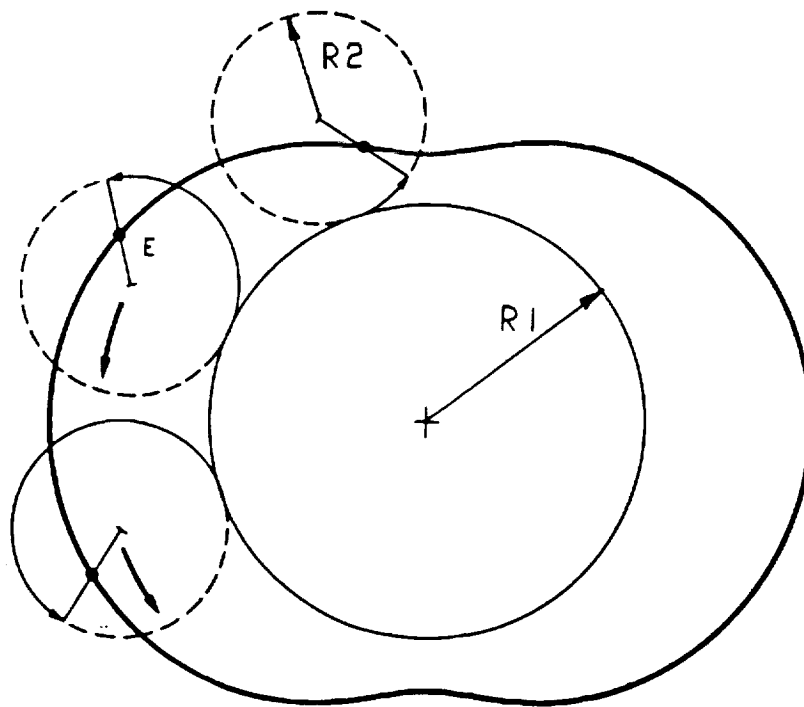


Figure 1.7: Illustration of epitrochoid generation.

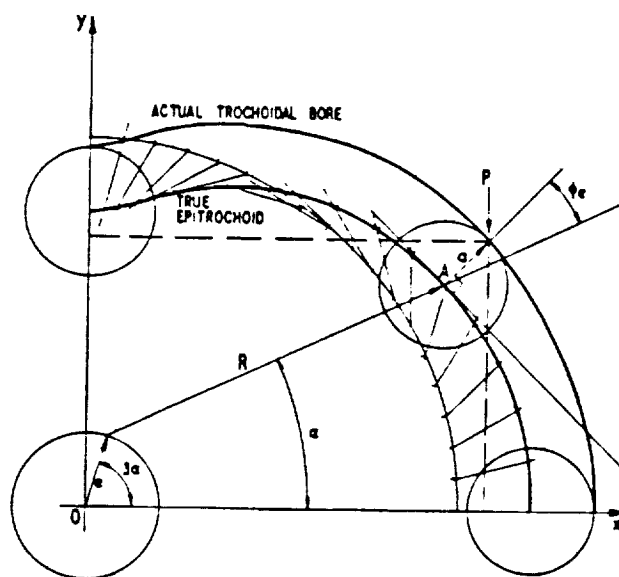


Figure 1.8: Illustration of trochoid generation..

If the true epitrochoid shape were utilized in the Wankel RCE, rotation of the rotor would force the seals to move in and out of their respective slots. Therefore, the trochoidal shape, shown in figure 1.8, is employed. By introducing the parameter (a), the translation between the apex seal and its containing slot is eliminated.

## 2.00 MODEL EXPECTATIONS

In determining how the finite element model (FEM) should be constructed, certain parameters were considered. Parameters such as model expectations, housing response, modeling accuracy, and result interpretation were defined and evaluated. With the optimum mix of these parameters, the FEM was constructed and tested to see if it accurately predicted the actual response to specified inputs in an efficient manner. The methods used in constructing the FEM and the manner in which they were verified are presented in this thesis.

Looking at the housing's geometry (see figure 1.4) and assuming the loading due to running conditions will be some type of fluctuating thermal and pressure loads, several conditions prevail. First, the inner and outer surfaces experience a complex dynamic loading which includes bending, membrane, and thermal stresses. Second, the primary ribs experience a simple fluctuating compressive load. Third, the spark plug and exhaust port regions are high thermal stress areas. Fourth and finally, no symmetry considerations are available since the endplates do not exhibit similar contact forces on either side of the housing.

From reviewing these conditions and the engine configuration, it is obvious that construction of a FEM that is capable of performing a dynamic analysis of the center housing under running conditions is far beyond the scope of a single thesis. Therefore, as an initial thrust toward a FEM fully capable of performing a dynamic analysis of the housing, a FEM was constructed and a static deformation analysis was performed. Furthermore, verification of the FEM was completed by utilizing the results from experimental testing of the actual housing. However, since the FEM of the housing is to be a base on which further study will be done it was built with the capability of

completing all the required studies: thermal, dynamic, and endplate interaction, with only minor changes. So, though only a static analysis was done, the construction of the FEM was completed with the running conditions previously mentioned in mind.

## 2.10 MODELING REQUIREMENTS

With no previous studies to draw from, the initial thoughts on modeling the housing were to construct it as two elongated cylinders with solid connecting ribs. Presumably, the cylinders could be modeled with basic shell elements and the ribs with basic 3-D solid "brick" elements. The additional geometry, exhaust and intake ports, spark plug, secondary ribs, and mid-plane stiffeners could be constructed using various types of shell elements. Some added qualities, though, would be required for the elements in certain regions. For example, thermal stresses would be high in the exhaust port and spark plug areas so the elements modeling those areas would need the additional capability of modeling temperature distributions. The inner and outer surfaces would also require this capability. The structural characteristics of the housing would require the elements modeling the inner and outer surfaces to be able to model bending, membrane, and shearing actions. In the primary ribs and port regions, though, elements would undergo only very simple structural loading. Furthermore, the elements modeling the stiffening agents would be required to stiffen the area involved in the most accurate manner possible. These considerations, being just an initial estimate of what is required to model the housing, were built and expanded upon as the study proceeded.

## 2.20 MODELING CAPABILITIES

During the study, several finite element codes were

available: SAP6, SUPERB, and NASTRAN. But due to the complicated geometry, a graphical pre- and post-processor was required. SUPERTAB was, for most of the study, the only pre-processor available and was only interactive with SUPERB. Since the need for a more general FEM program such as NASTRAN or ANSYS was not great and the need for a graphical pre-processor was, SUPE was chosen to solve the model and its supporting tests.

SUPERTAB creates the specific geometry using points, lines, edges, surfaces, and volumes. Meshes are placed on the defined geometry then nodes and elements are created according to the mesh size and configuration. The SUPERTAB file is then adjusted into SUPERB format. Using the nodal coordinates and element connectivity defined in this file, SUPERB solves the set of simultaneous equations obtained with an elemental wavefront technique. The wavefront of a finite element model is the number of degrees of freedom needed to be held by the computer in order to solve for a specific displacement. In general, an elemental wavefront solution differs from a nodal bandwidth solution in that a relatively small main frame computer is needed but an extreme amount of storage space is required.

When constructing geometry with SUPERTAB, a prescribed order must be followed (see figure 2.1). Points must be defined before lines, arcs, and splines can be created. Edges need to be defined before surfaces or volumes can be created. Also, the order in which construction occurs determines a number of key parameters. Nodal and elemental numbering, coordinate system orientation, and in the case of thick shell elements, actual element configurations are determined by the sequence of detecting existing pieces of geometry.

When defining the elements to be used, the specific element type must be selected. SUPERB has a limited finite element library but does support the basic types capable of modeling the housing. Interestingly enough, SUPERB supports two variations of shell elements: a 2-D thin shell and a 3-D thick shell element. Both of these shell elements have bending, membrane, and shearing



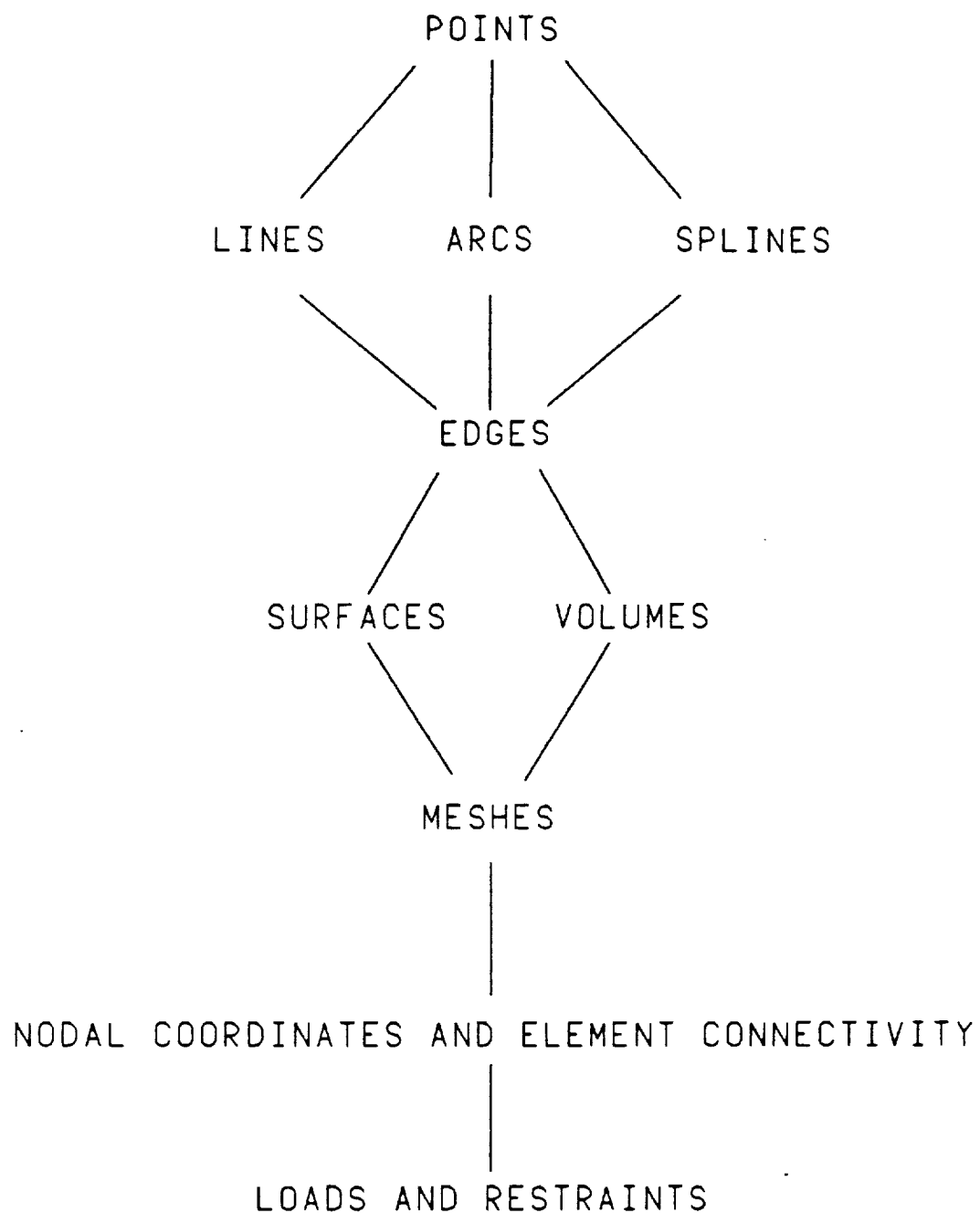


FIGURE 2.1: SUPERTAB GENERATION LEVELS.

capabilities. Also, both elements are capable of predicting in-plane temperature distributions, but only the thick shell element is capable of modeling a temperature distribution through the shell's thickness. Furthermore, each shell element is available in various nodal configurations. A more indepth discussion on each element type is listed in appendix 1.

Since two types of shell elements were available to model the complex and varied loading conditions, a benchmark study was completed. The study was designed to determine the qualities and performance of each shell element. Mesh sizes, aspect ratios, and nodal configurations were varied and the results analyzed. The actual tests and results are given in appendix 2. A basic summary of this study, though, reveals four important shell element characteristics:

- 1) Linear (4-node) thin shell elements are not capable of modeling shearing actions and therefore should not be used in areas where shearing actions are evident.
- 2) Linear (8-node) thick shell elements perform very well but converge slowly so a smaller mesh size is required.
- 3) Parabolic (8-node) thin and (16-node) thick shell elements perform extremely well and exactly alike.
- 4) The smallest recommended mesh size for modeling the housing would require at least six elements in the axial direction. A reduction to four elements would produce an estimated 10% error.

These attributes, along with the considerations mentioned in appendix 1 provided the necessary knowledge to begin modeling the housing.

## 2.30 MODEL CONSTRUCTION

With the necessary information on hand, actual model construction began. Proceeding with the assumption that the housing could be constructed using a mix of solid and shell

elements, ideas on how and where these elements would be used were discussed. Since a complex stress field was not expected, solid elements would be used exclusively in modeling the primary ribs. Furthermore, due to the high temperatures expected, thick shell elements would be required in the port and spark plug regions. For the remaining geometry: the inner and outer surfaces, secondary ribs and mid-plane stiffeners, the pros and cons of utilizing either thick or thin shell element were weighed. The thin shell element was simple to construct, contained fewer nodes than the thick shell element, and performed extremely well. However, there were difficulties in implementing the thin shell elements. For one thing, a compatibility problem was introduced whenever thin shell and solid elements would be joined. Also, the thin shell element assumed a constant thermal distribution through its thickness and, due to its construction, only the displacement of its middle surface was available. On the other hand, the thick shell element had excellent thermal as well as structural capabilities, portrayed the actual geometry shape, and had no compatibility problems with solid elements. It does contain twice as many nodes as the thin shell element but, since each node supports half the degrees of freedom, the model's size is not affected. Taking this into account, the thick shell element was determined to be an excellent choice to model the inner and outer surfaces. The secondary ribs and mid-plane stiffeners, however, would be modeled with thin shell elements. This was because the adverse qualities of the thin shell element did not affect its performance in these regions.

The final aspect of the initial construction phase was that the model had to be constructed utilizing parabolic nodal configurations. The performance of the shell elements were greatly affected by this parameter. Furthermore, even though the solid elements performance was not affected by its nodal configuration, they also would utilize parabolic elements. This would ease construction since the use of transition elements would not be required.

Given the blueprint for the final model, outlined above,

convergence requirements as well as computer limitations were considered. In order to satisfy both, a modeling sequence was established. Modeling construction methods as well as the model's capabilities were improved and enhanced as determined by previous models. Basically, the sequence consisted of four models: the linear thin shell model, a parabolic thin shell model, a parabolic thick shell model, and the final model which utilized all of the proven construction features. The actual construction and the subsequent analysis of each model is presented later on in this thesis.

### 3.00 CENTER HOUSE TESTING

After the construction sequence was determined and implemented, methods for verification of each model were considered. Undoubtedly, experimental tests of the actual housing were necessary. But, it was important that the testing be indicative of the dynamic stress fields the housing would encounter during operation. Thus, a test which applied a stress field that encompassed bending, membrane, shearing, and uniaxial effects in varying degrees was required. To accomplish this, two tests were performed. The initial test consisted of merely subjecting the housing to an internal pressure. However, due to complications in simulating this test, a second test was performed. This second test, a simple tensile test, was easily simulated and provided a loading that exercised each model's capability to predict a complex stress field.

#### 3.10 UNIFORM INTERNAL PRESSURE TEST (IPT)

The first test was performed by NASA. The housing was sealed and pressurized to 500 psi in increments of 100 psi. At each increment the strains in three locations were recorded. This was accomplished through the use of strain gage rosettes mounted by NASA. A graphical representation of the test and the location of the rosettes is given in figure 3.1. These strains are plotted in figures 3.2-3.4.

Several tests were performed and consistent data was obtained. Yet, in analyzing the data, certain discrepancies were noticed. First, the strains were scattered. Second, the maximum principal stress occurred in the axial direction instead of the

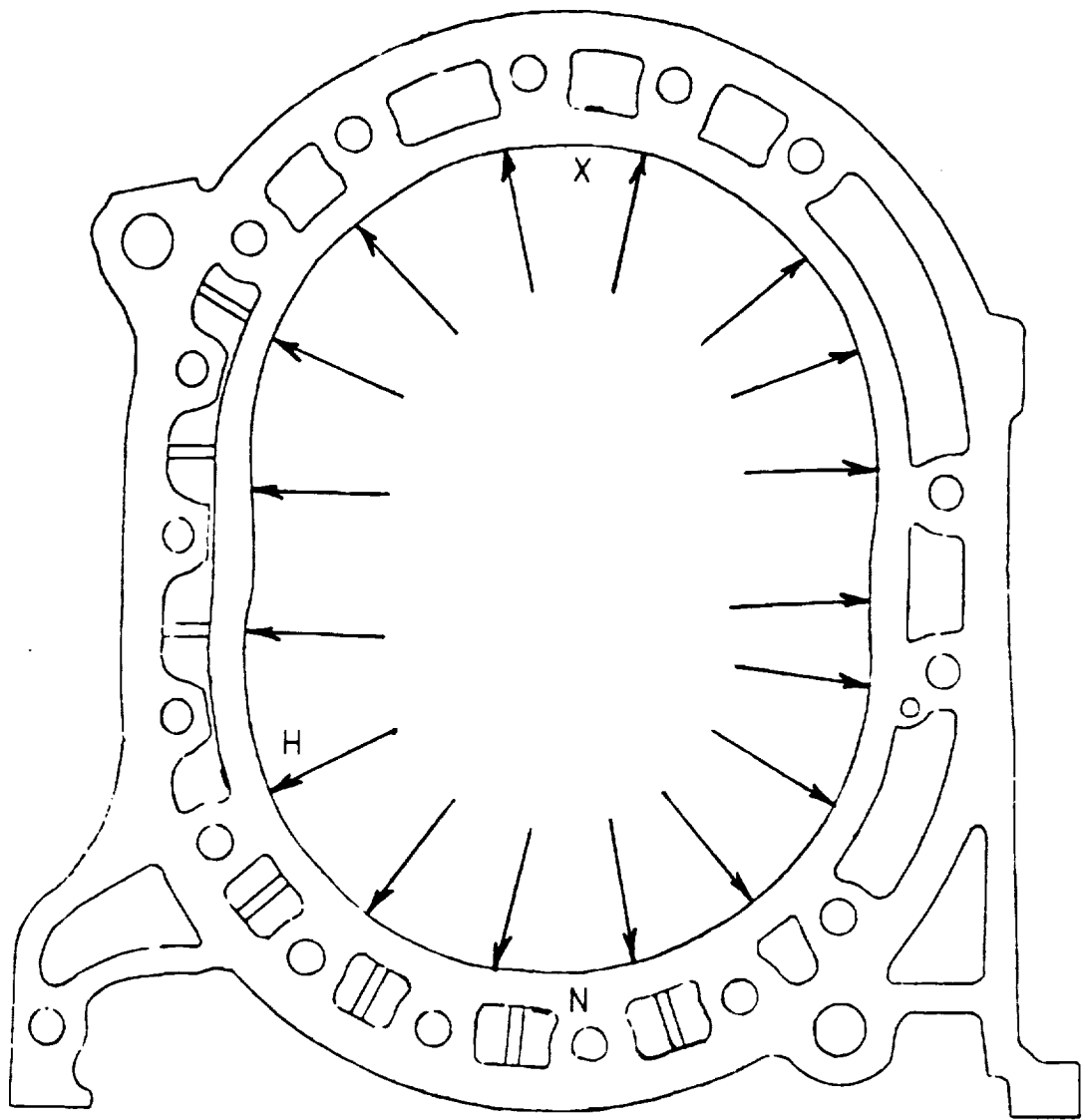


Figure 3.1: Illustration of the internal pressure test and locations of the strain gage rosettes.

location n

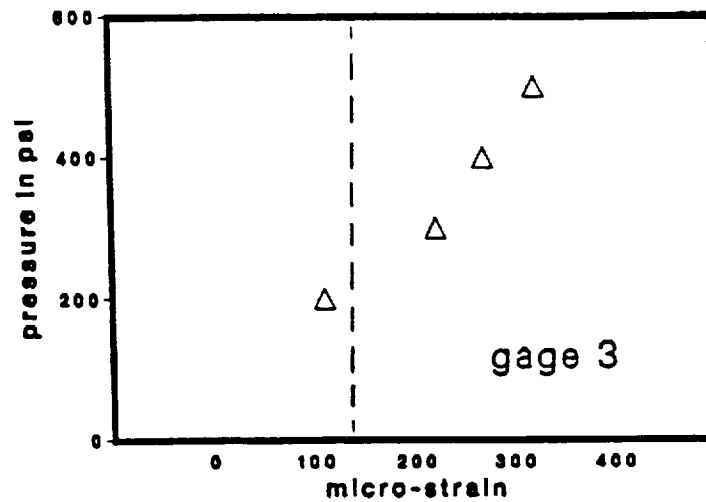
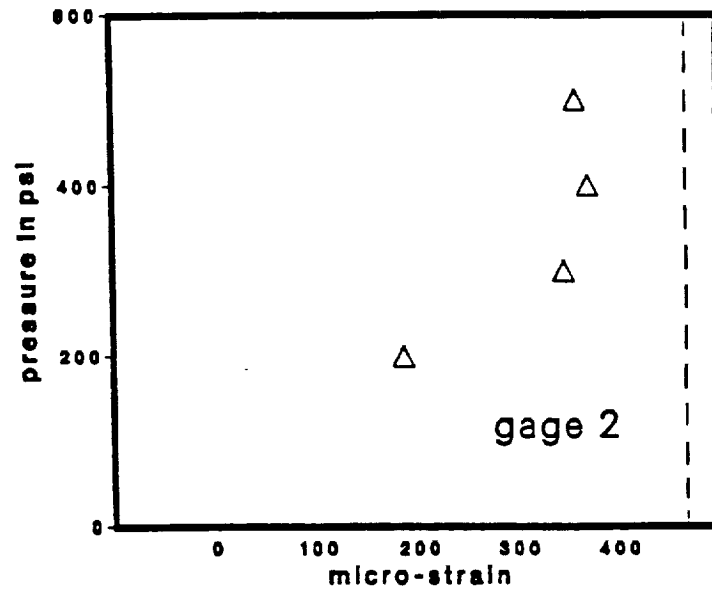
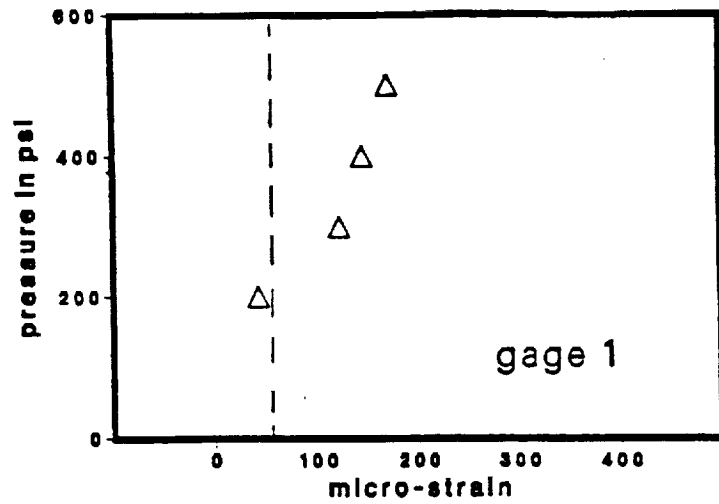


figure 3.2; internal pressure test.

location x

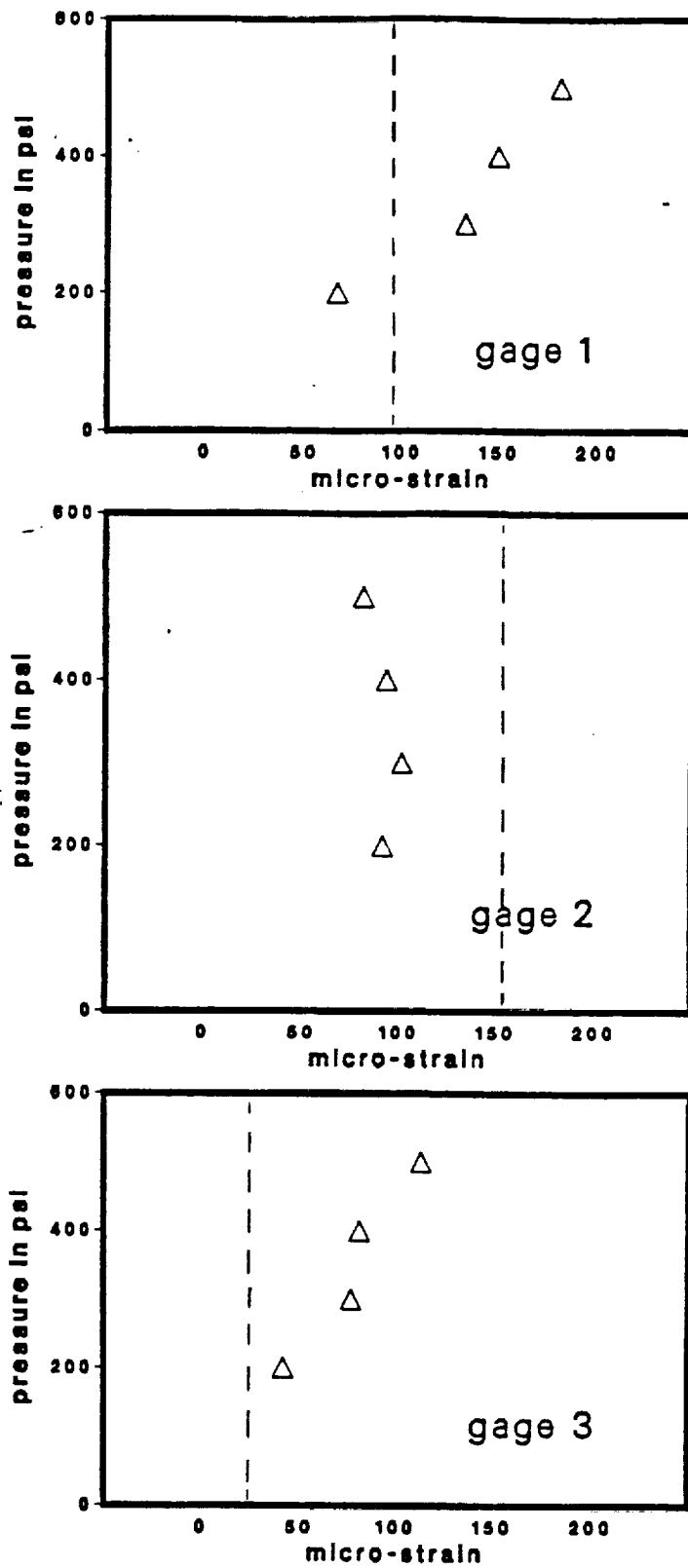


figure 3.4; internal pressure test.



# location h

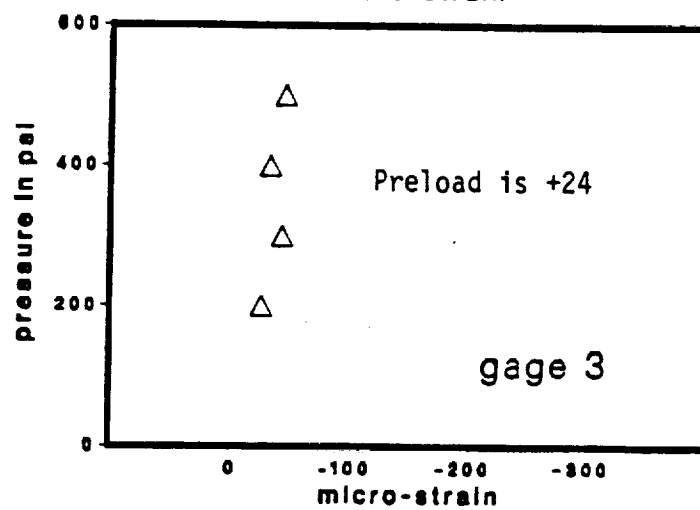
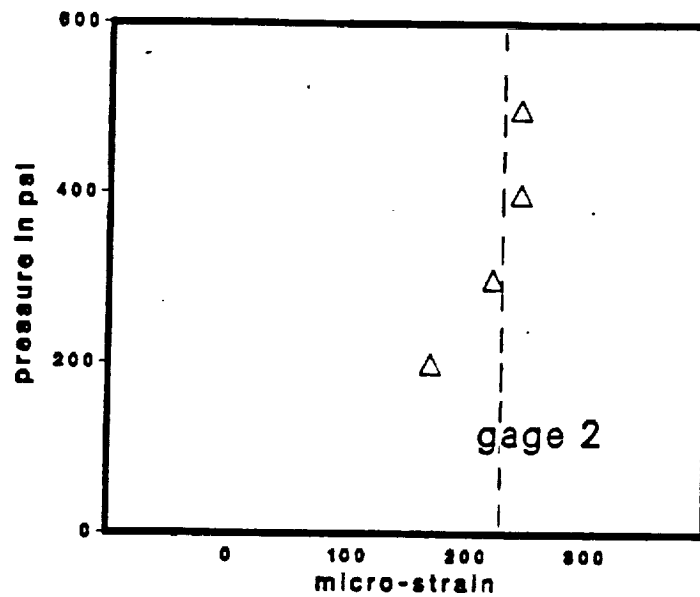
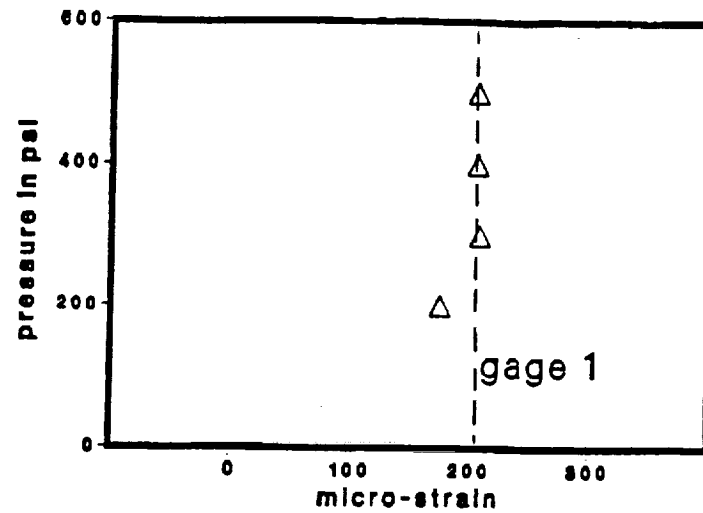


figure 3.3; internal pressure test.

hoop direction. Third and finally, the stress at location N, which is a heavily reinforced region, was computed to be larger than the stress at location H, which is a region with very little support. There were other puzzling features but these discrepancies alone forced further investigation into the test. Inevitably, this led to reviewing the test procedure and the manner in which it was executed.

The procedure was very straight forward. In order to pressurize the housing it needed to be sealed. This sealing was accomplished by bolting two aluminum plates to either side of the housing. After the bolts were tightened, the gages were zeroed and the housing was pressurized. Thus, the preload that initially existed on the housing from tightening the bolts had been neglected. When the housing was pressurized, the pressure proceeded to relieve the compressive load which induced a tensile strain in the gages. The summation of the preload relief and the pressurization was what the gages actually read. In an attempt to grasp the effect of the preload, the bolts were loosened and resulting strains recorded. The vertical line on graphs 3.2-3.4 indicate these strains. As can be seen, the effect of the preload varied throughout the housing.

Theories were hypothesized on what was occurring within the housing. Summarized, these theories were that either the housing deformed until the clearance in the bolts was taken up, or, the preload was completely relieved and deformation of the endplates, simulated in figure 3.5, was occurring. Although this occurrence needed to be studied, it was not easily simulated and study of it was not proposed by this thesis. Therefore, a second test was needed which could be easily simulated and would also verify each model.

### 3.20 DISCUSSION OF THE TENSILE TEST

Due to the unpredictable parameters of the internal pressure test, an additional test was required. Realizing the undesirable

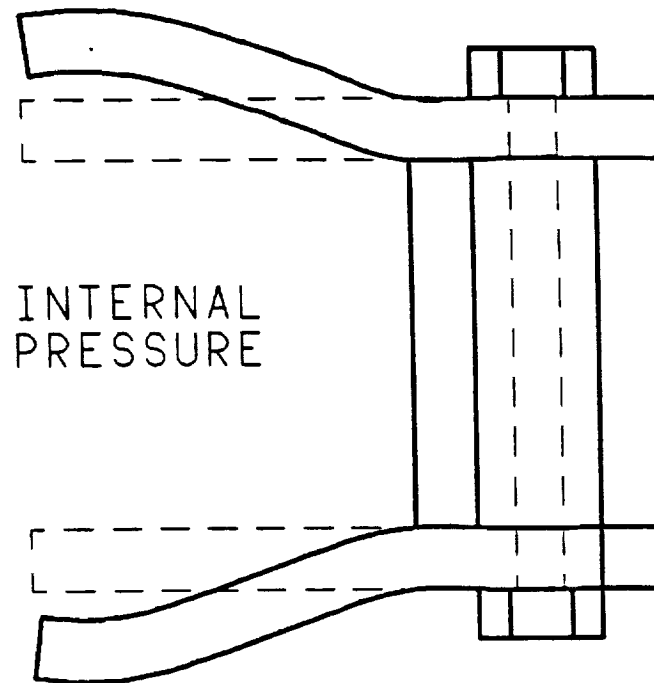


Figure 3.5: Simulated endplate deformation.

features of the first test, the second test had to not only be indicative of the conditions due to operation but also it had to be easily simulated. These two restrictions were satisfied by a simple tensile test. Even though the test would not actually simulate pressurization, the stresses incurred would be similar in various regions. Combinations of bending, membrane, and shearing actions would stress the model much like operating conditions.

To perform this test, the housing was obtained from NASA. Two devices were made (see figure 3.6) which enabled the housing to be tested using the University's Tinius Olsen testing machine. As seen in figures 3.7 and 3.8, the housing was loaded in two directions. Results were obtained by using the strain gage rosettes mounted by NASA. Loads were incremented until enough strain (15 micro) was acquired to overcome any errors involving gage sensitivity. By carefully monitoring each rosette, permanent deformation was avoided. Also, to aid in simulation, the FEM's global axes were placed on the housing.

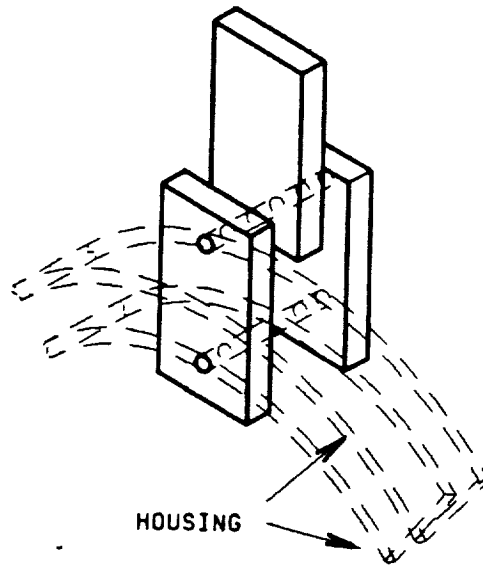


Figure 3.6: Illustration of devises used to test housing.

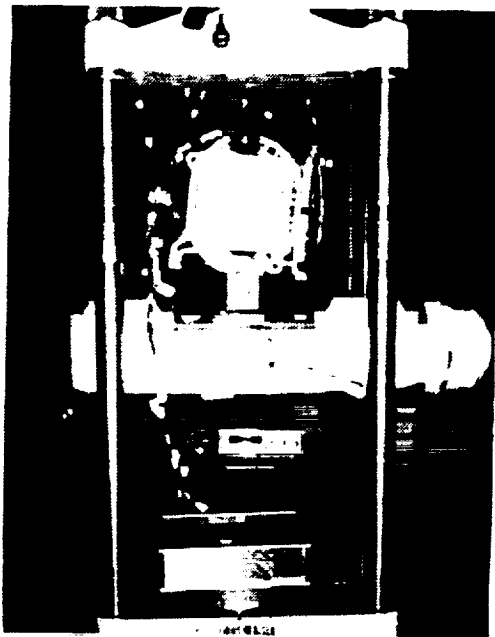


Figure 3.7  
Longitudinal tensile test.

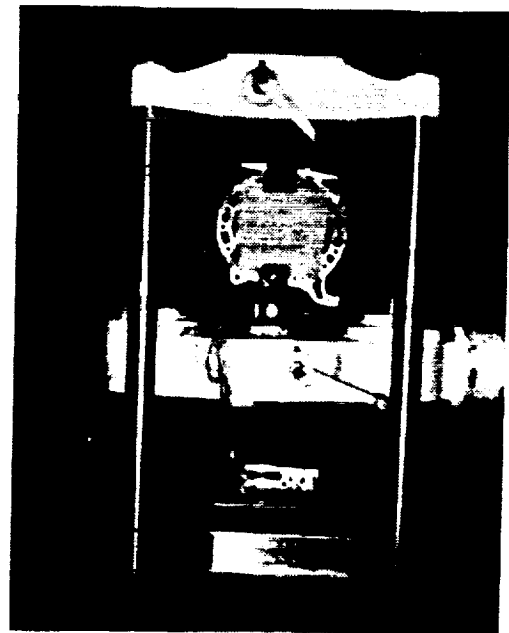


Figure 3.8  
Transverse tensile test.

Using this as a reference, the loading directions were easily obtained.

The data obtained from performing the tensile test was very consistent. Strains recorded were, as seen in figures 3.9-3.11 and 3.12-3.14, linear. The strains recorded for the tensile transverse test are reported, but due to inconsistencies, they were not relied upon for model verification. Furthermore, the stresses resulting from these strains were computed and are listed in table 1. In this table, principal stresses and directions are given, but also stresses are given in the model's expected coordinate system explained in appendix 1.

### 3.30 SIMULATION OF THE TENSILE TESTS

Once the results of both tensile tests were recorded and analyzed, simulation of the tests was completed. The orientation of the global axes during the test was known. From this information the longitudinal loading of each model was directed 5 degrees from the vertical and a load of substantial magnitude, 1000 pounds, was applied to each model. Since clearance in the bolts was large, loading caused the bolts to deform (simulated in figure 3.15). Thus, distribution of the load varied. Deformation of the bolts applied the load symmetrically across the bolt hole's length, but the effect of the added deformation could not be fully modeled. A load distributed across the bolt hole's length was utilized for verification but, in order to bracket the actual load application, two other loading distributions were investigated. One, a single point load located in the center of the bolt hole's length, and a second, which applied equal loads on either end of the bolt hole. The actual loading distribution was expected to occur between these two extremes.

To run the test, each model needed to be supported. However,

# location n

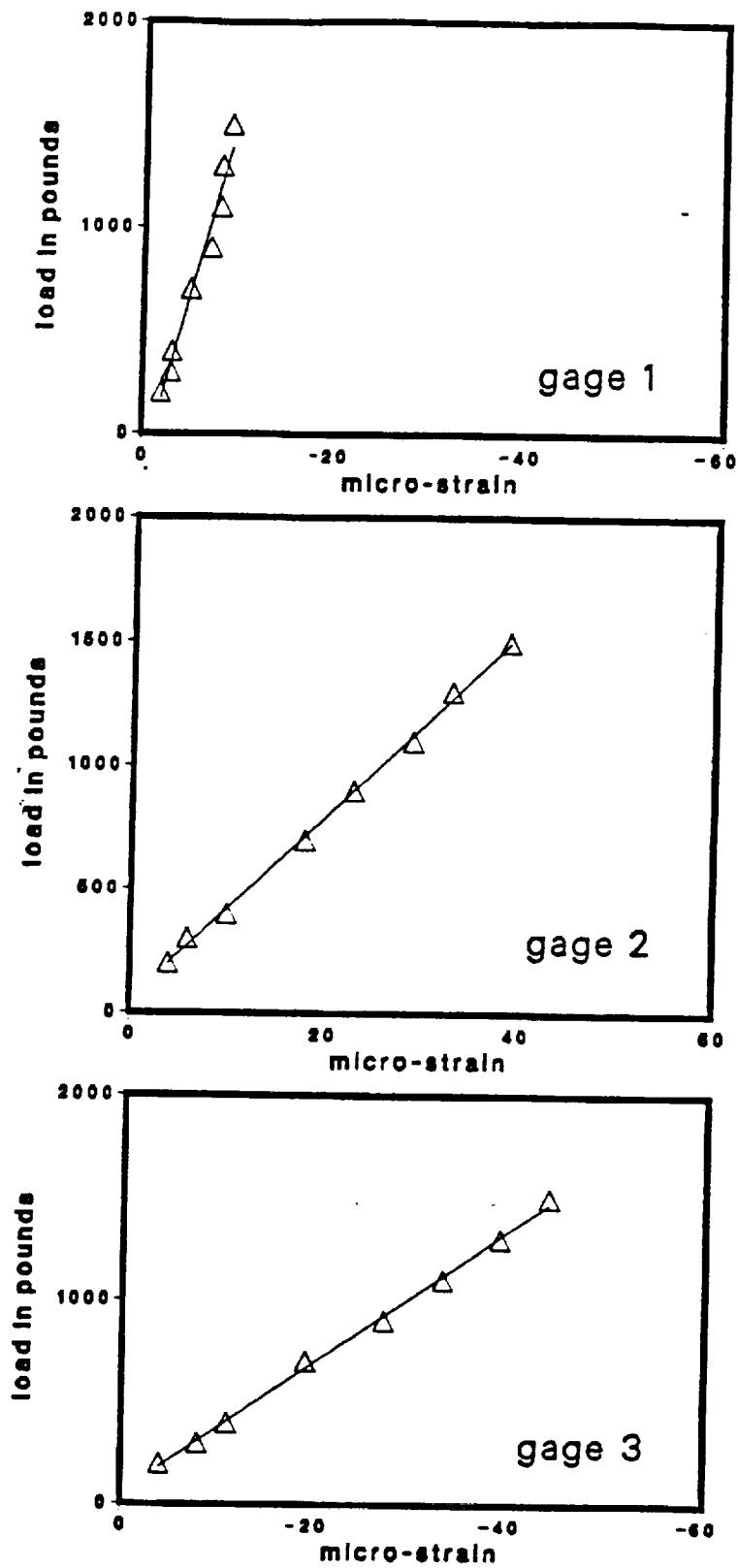


figure 3.9; longitudinal tensile test.

# location h

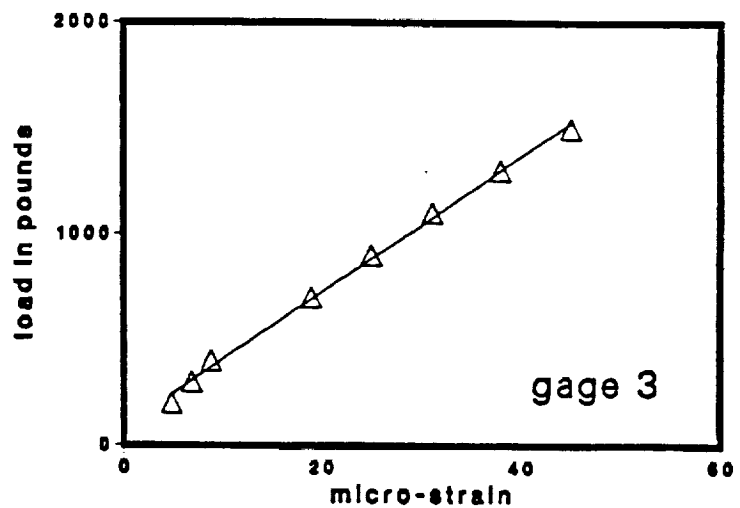
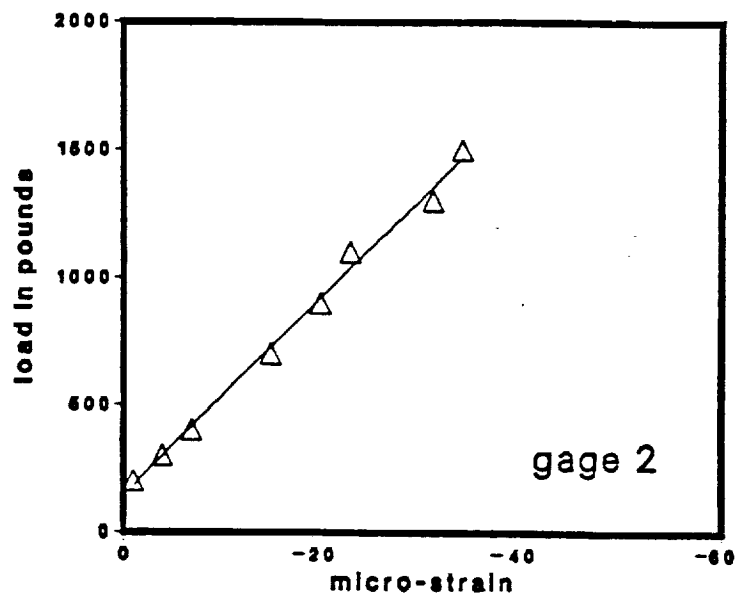
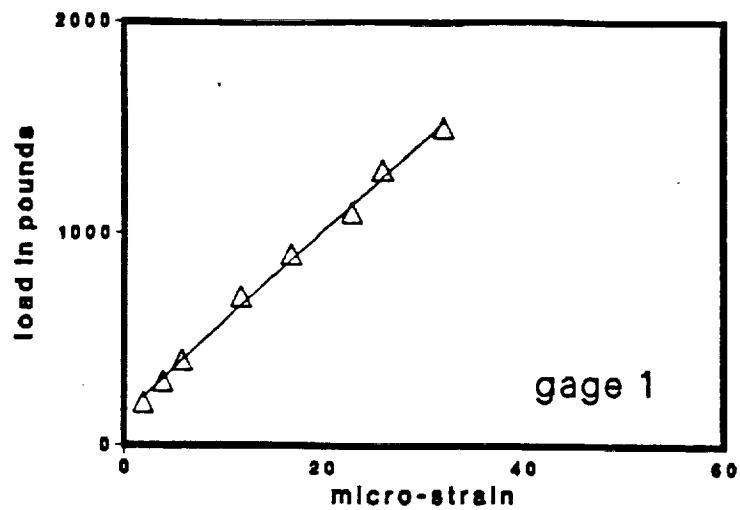


figure 3.10; longitudinal tensile test.

location x

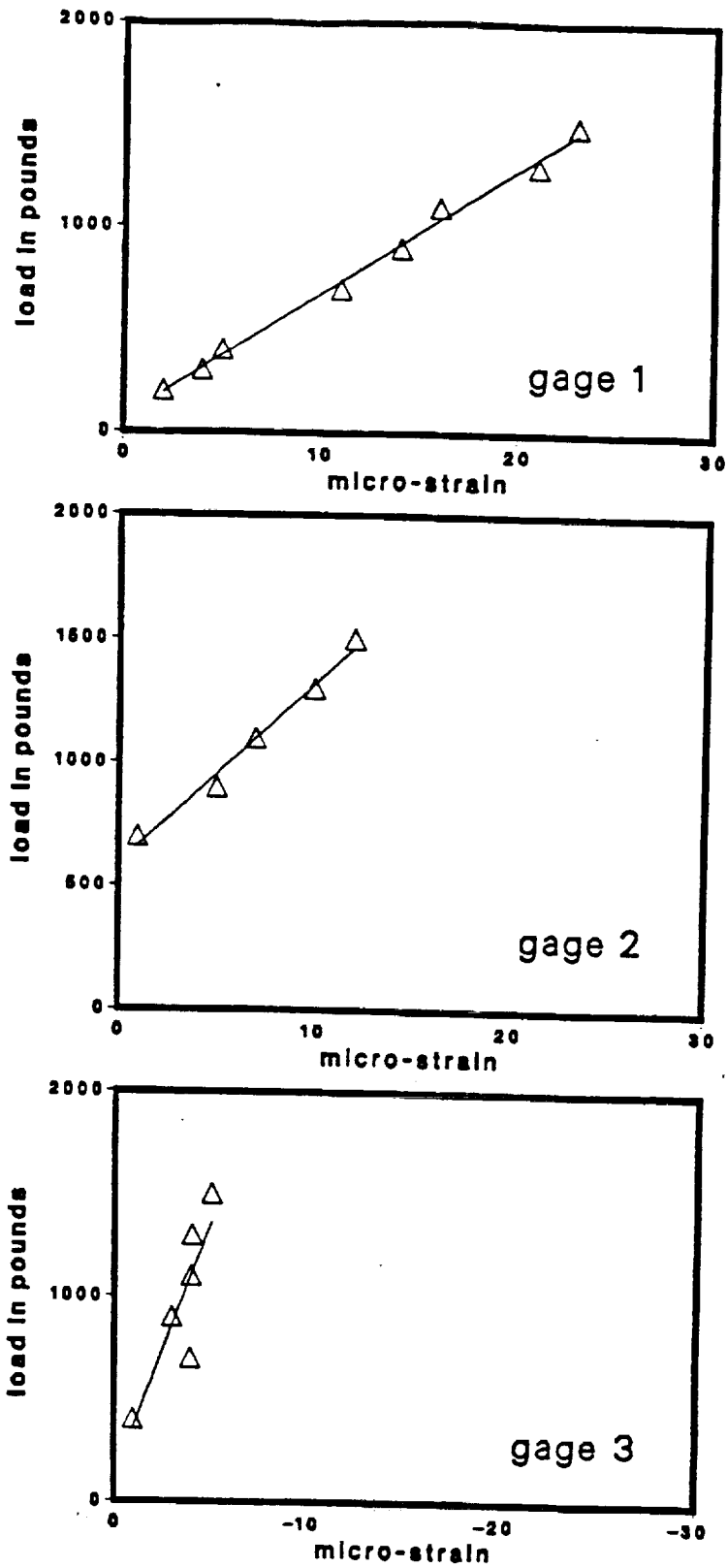


figure 3.11; longitudinal tensile test.



location n

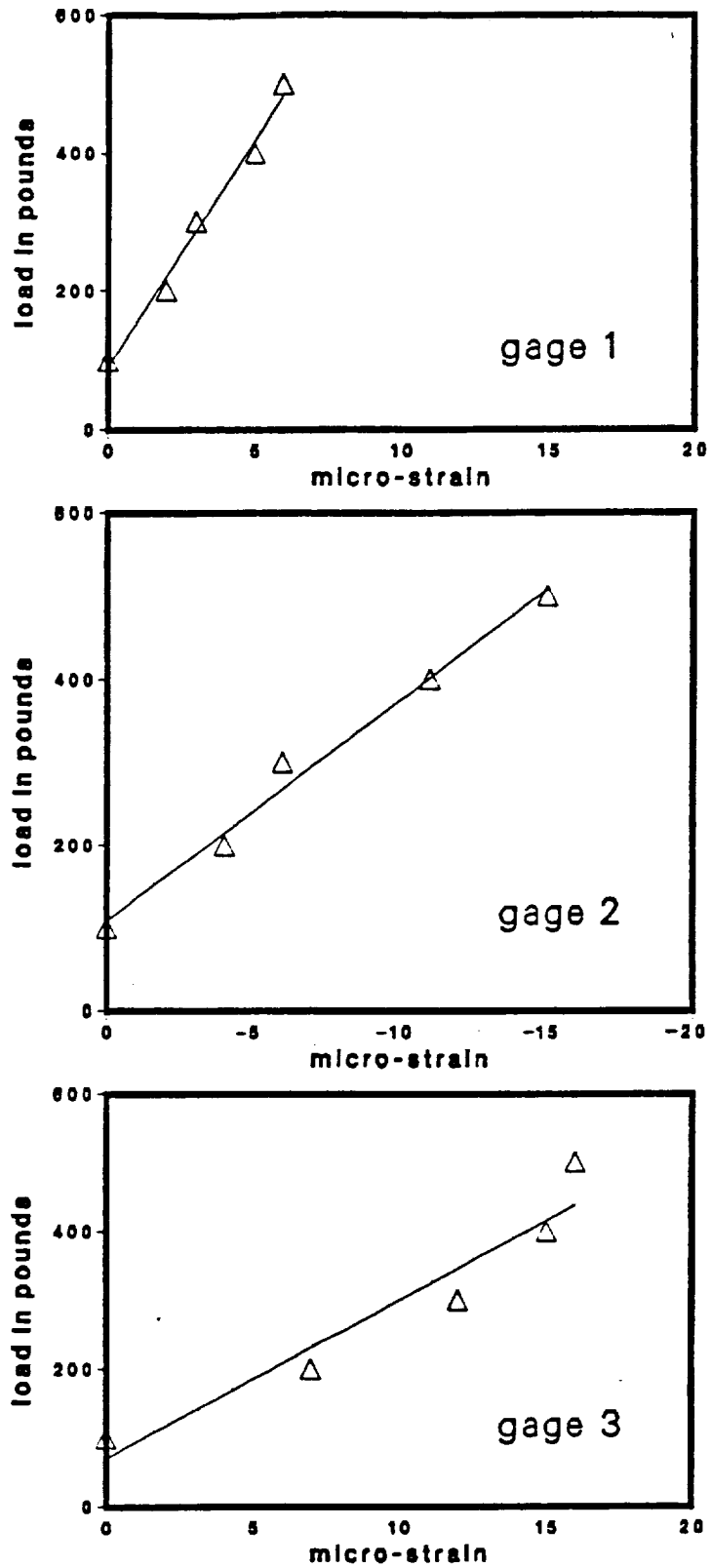


figure 3.12; transverse tensile test.

# location h

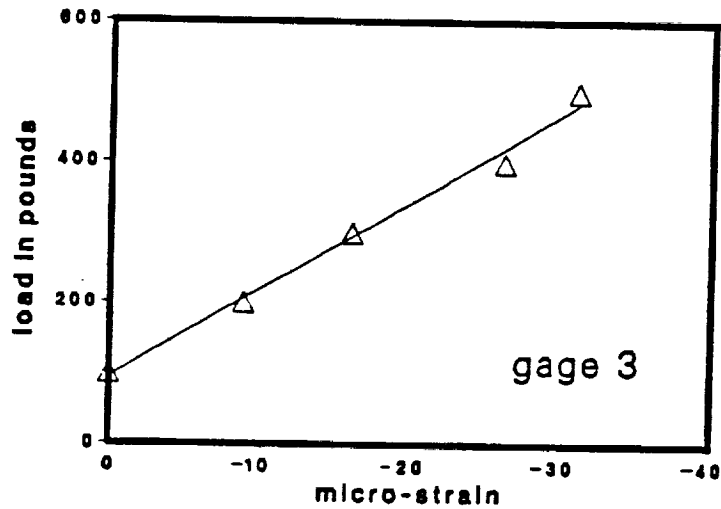
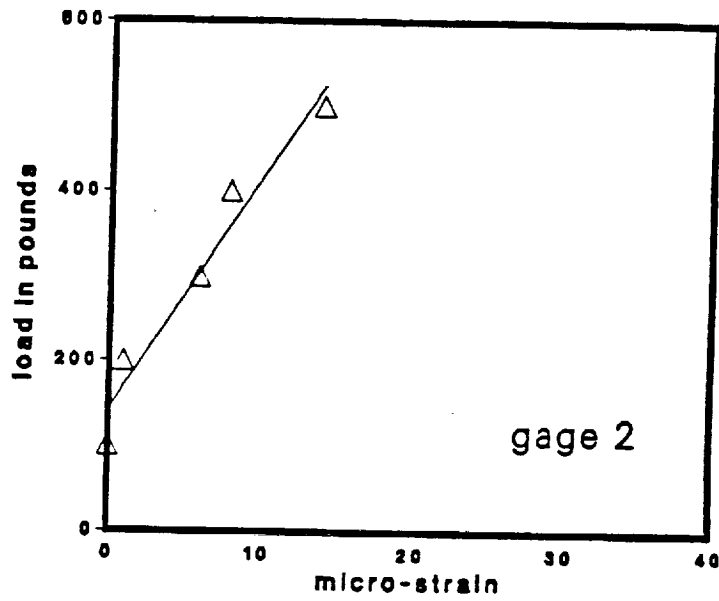
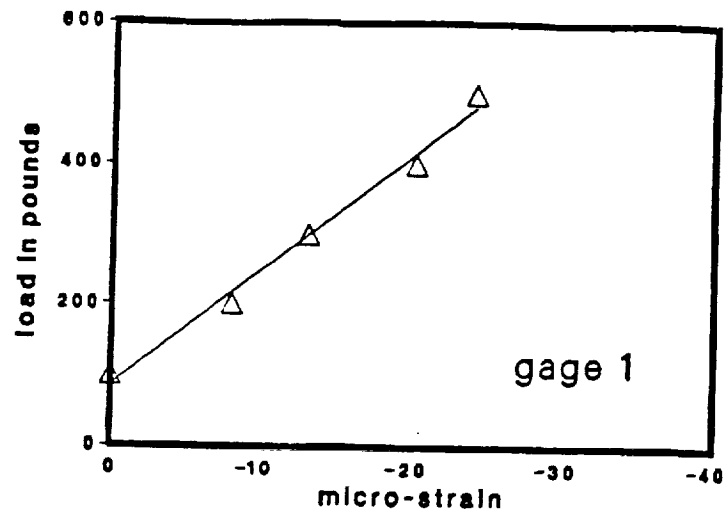


figure 3.13; transverse tensile test.

location x

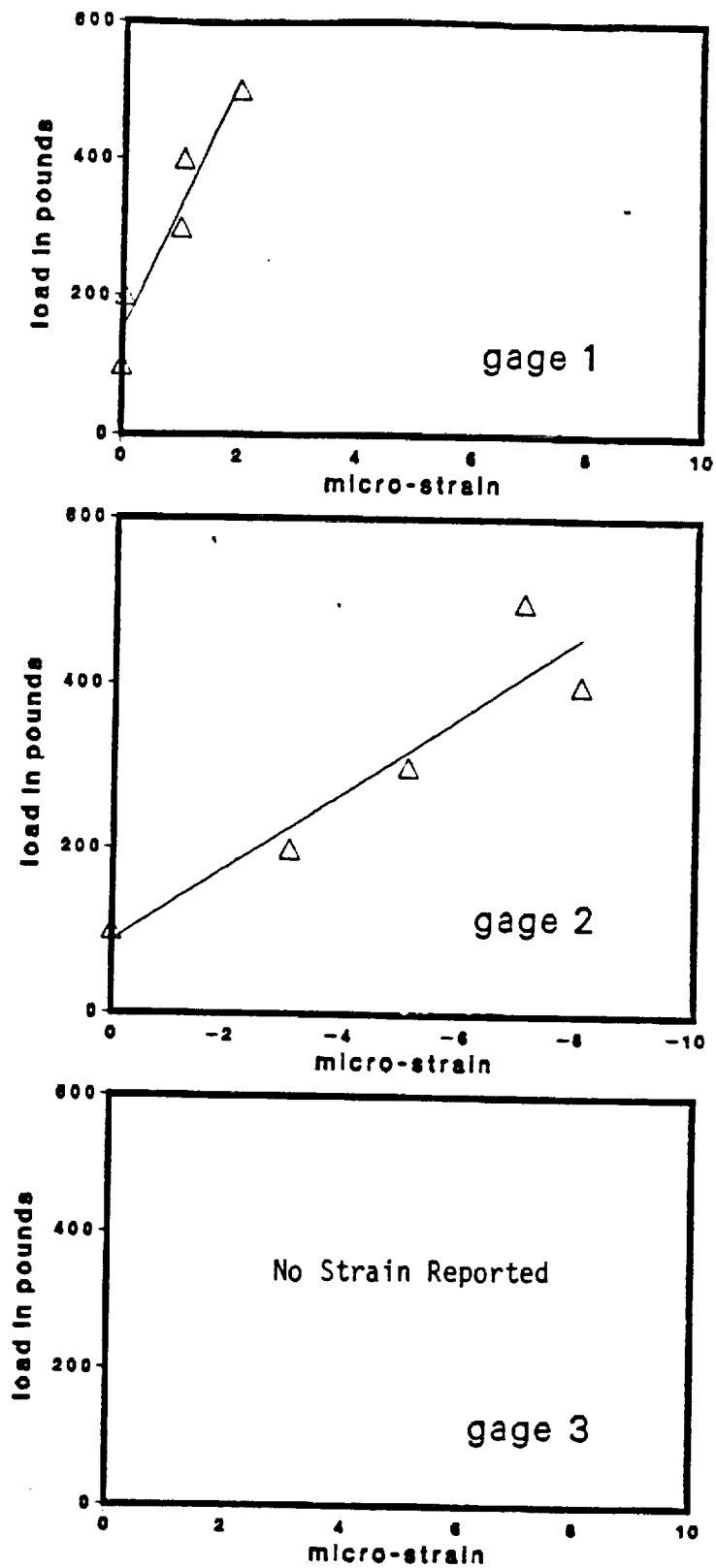


figure 3.14; transverse tensile test.

LOCATION	N	H	X
SIGMA-X	75	-9.3	85
SIGMA-Y	-597	702	100
TAU-XY	-85	36	-76
SIGMA-P1	86	704	170
SIGMA-P2	-607	-11	15
DIRECTION	7.1	2.9	47

\*NOTE: POSITIVE DIRECTION INDICATES CLOCKWISE ROTATION OF AXES.

STRESSES GIVEN IN PSI

Table 1: Stresses from longitudinal tensile test (1000 lbs.).

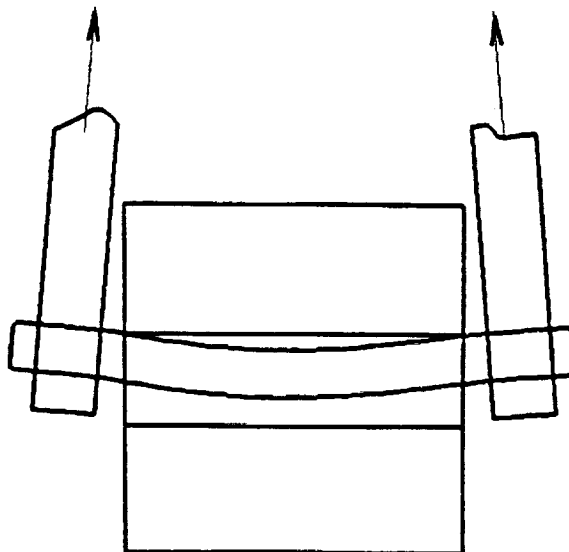


figure 3.15: Illustration of suspected bolt deformation.

support of the model in the regions around the rosette locations would affect the validity of the results. Consequently, support of the model was restricted to being located away from the gage locations, as shown in figure 3.16. In addition, actual support of the model had to inhibit the model as little as possible. Although two support systems, shown in figure 3.17, were utilized, various support systems were looked into. The first type was used primarily in the thin shell models as shown in figure 3.17a. It restrained all the degrees of freedom of one node. The remaining nodes in the vertical row were able to translate vertically and only the vertical rotation was constrained. This scheme enabled the model to deform in the most natural manner possible. The second type, shown in figure 3.17b, was designed for the thick shell models since nodal rotations in the models could not be restrained. In order to restrict the rotation of the model about its vertical axis, two rows of nodes were restrained.

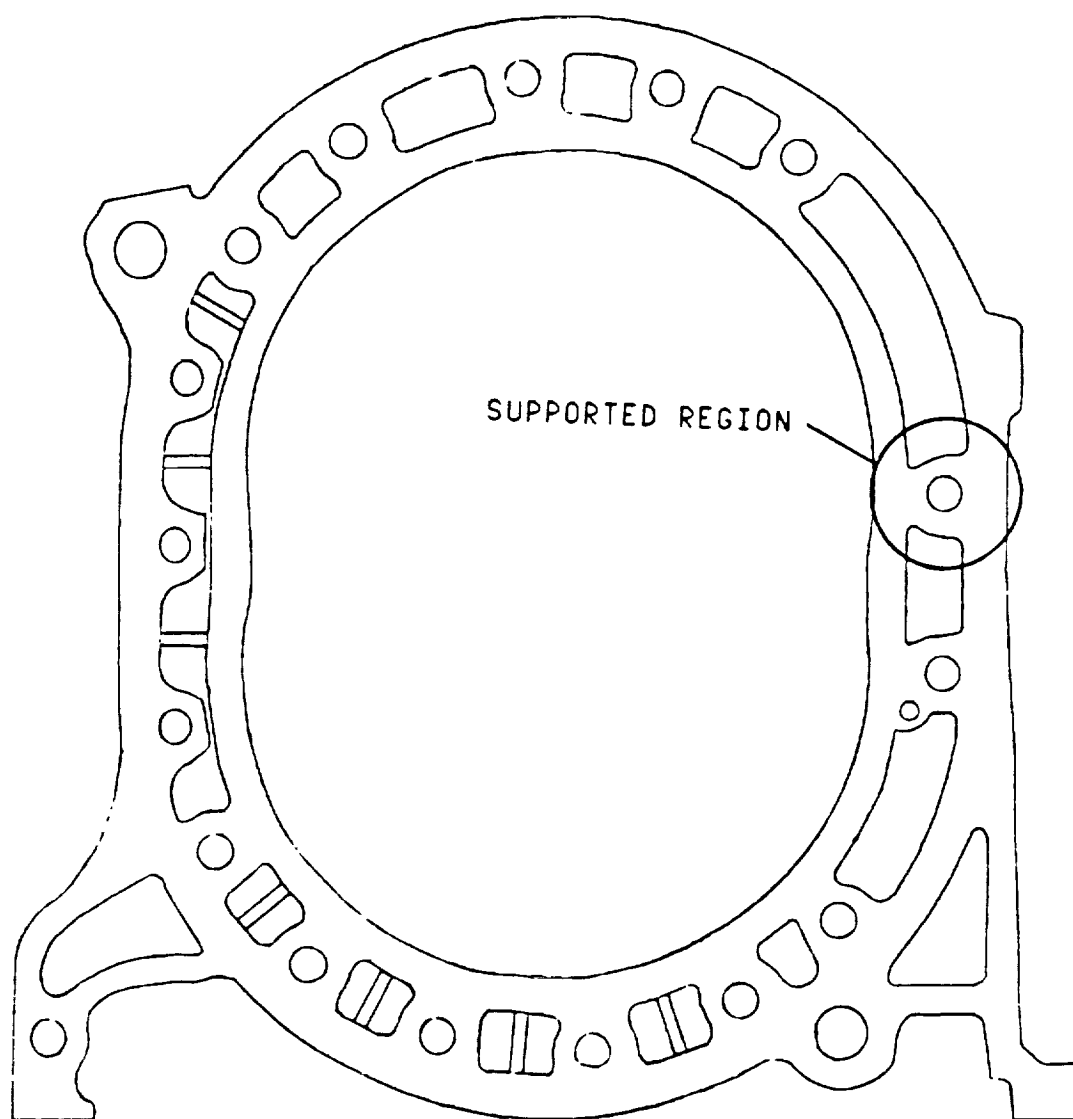
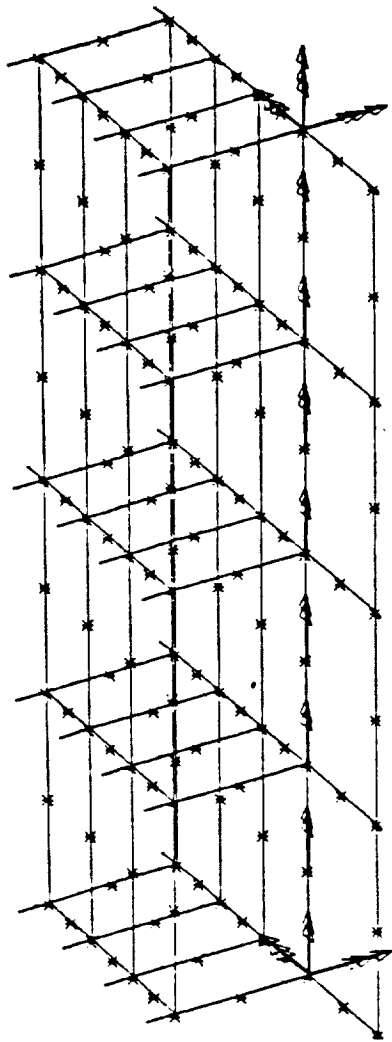
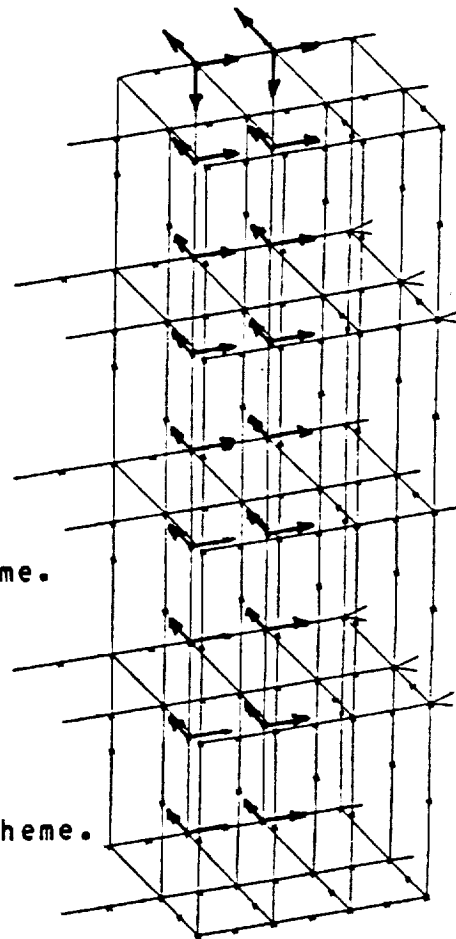


Figure 3.16: Location of model support.



(a) Thin shell model's support scheme.



(b) Thick shell model's support scheme.

Figure 3.17: Illustration of support schemes for thin and thick shell models.

#### 4.00 CONSTRUCTION METHODS AND MODEL VALIDITY

As mentioned earlier, the construction of the FEM of the housing was done with a series of models. Each model in the series utilized construction methods proven by the previous models. Initially, the sequence was to consist of four models: a linear thin shell, parabolic thin shell, parabolic thick shell, and the final model. However, as expected, modifications to this plan were required. Mesh sizes, element types, and nodal configurations were methodically modified until the final model, vaguely introduced in chapter 2, was constructed. By utilizing this sequence, convergence was checked and overall confidence in the model was gained. Thus, when it was completed, the final FEM not only accurately simulated the housing, but it also possessed a firm modeling foundation. In the next segments the construction methods and results of each model in the sequence are presented.

#### 4.10 CONSTRUCTION OF THE LINEAR THIN SHELL MODEL

As the first model of the construction sequence, the linear thin shell model was to be a simple FEM of the housing. Since simplicity was to be its greatest attribute, the model was constructed using elements with linear nodal configurations. This made the model's size a modest 9200 degrees of freedom. Furthermore, as seen in figure 4.1, the actual geometry construction was not complicated. Thin shell elements were utilized for the inner and outer shells, thus, only the middle surface of either shell needed to be defined. In addition, certain pieces of geometry were not included. Since it was believed that the regions where the strain gage rosettes were located would not be affected by the placement of the spark,



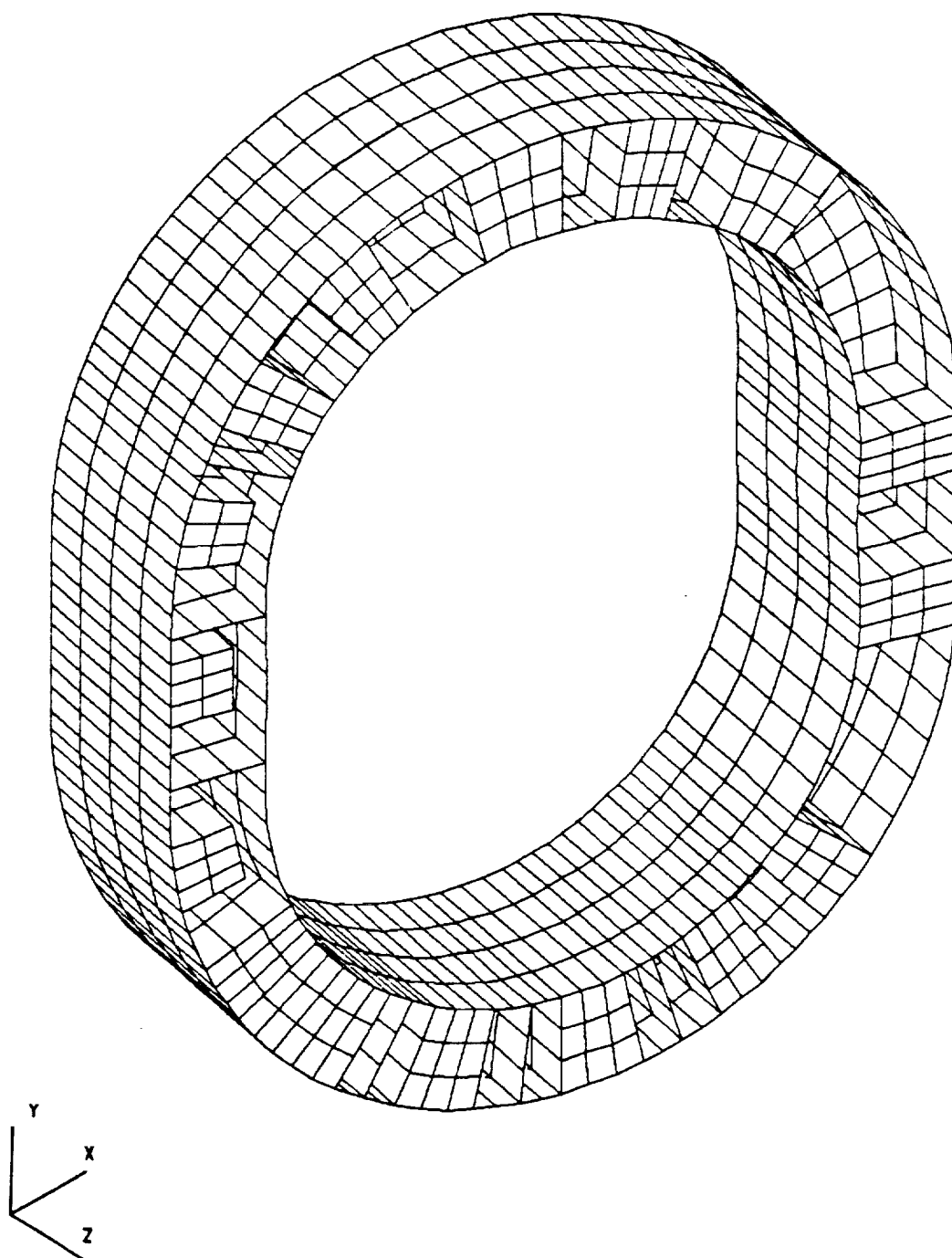


Figure 4.1: Illustration of linear thin shell model.

plugs, engine mounts, and both ports, this geometry was not modeled. Also, it should be noticed that the bolt holes were not modeled either. The significance of the bolt holes was considered to be negligible. However, in an endplate interaction study the significance of the bolt holes must be considered due to the tolerance of the bolts. Simplicity was further gained by using a mesh size that was uniform and considered coarse. Refinements of the mesh would be considered if necessary.

As construction of the linear thin shell model began, modeling methods materialized. Problems that arose were subsequently solved by employing new methods. Of the problems encountered, only one major difficulty stood out, this was compatibility between solid and thin shell elements. Since solid elements do not support nodal rotations, a compatibility problem was introduced whenever thin shell elements were to be modeled with solid elements. This particular problem occurred in two regions of the linear thin shell model. In these regions, the solution used was similar. Both utilized a method in which certain nodes were coupled together. By coupling these nodes, the translational degrees of freedom of the associated nodes were forced to move together. It must be noted here that the nodal rotations of the thin shell elements were not affected by this coupling. Thus, moments could not be translated across the elements in question. Although this method seemed adequate, it did not model the regions exactly, so doubts were raised on its accuracy.

Both regions in question were located around the primary ribs. The first being where the inner and outer shells connected to a primary rib. In order to model this particular region, two fully closed cylinders were constructed. The solid ribs were then placed between the two shells. By defining common meshes, coincident nodes were assured. These nodes were then coupled creating a "spot welded" effect. Figure 4.2 illustrates this construction technique.

In the second region, where the mid-plane stiffeners connect to primary ribs, the coupling method was used in much the same

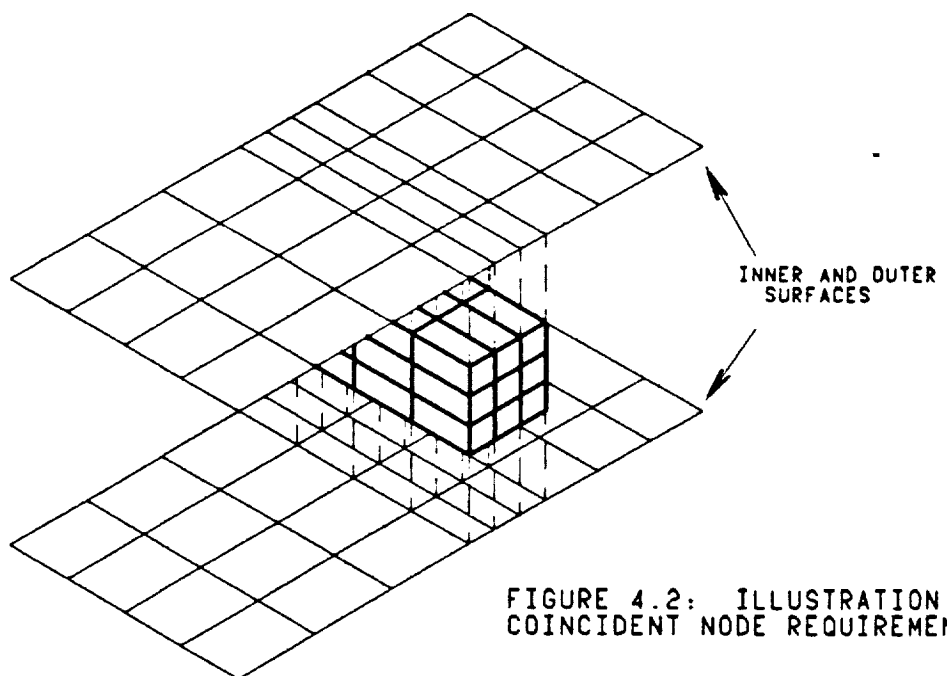


FIGURE 4.2: ILLUSTRATION OF COINCIDENT NODE REQUIREMENT.

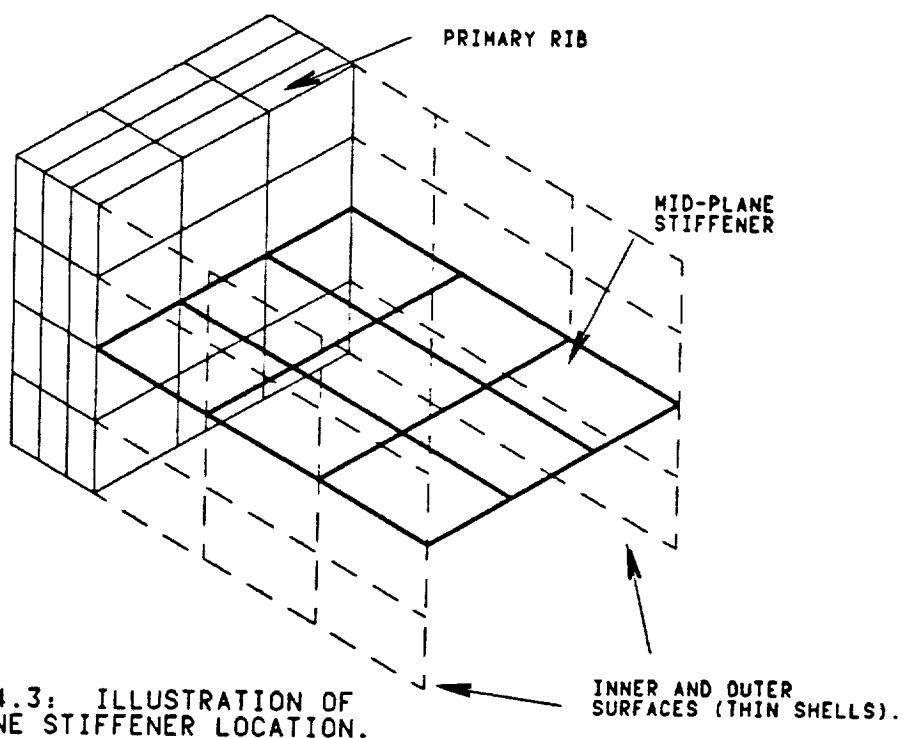


FIGURE 4.3: ILLUSTRATION OF MID-PLANE STIFFENER LOCATION.

manner. A common mesh produced coincident nodes which were then coupled. However, as can be seen in figure 4.3, the major support of this region originated from the intersection of the stiffener and the inner and outer shells.

As stated, the accuracy of these modeling techniques was doubted. Since rotations of the thin shell elements in these regions was possible, deformations as seen in figure 4.4, could occur. Yet, the effect of such deformation was not known. Clearly, this effect was negligible for the mid-plane stiffeners since the expected loading direction would not produce undue deformation. Only if a pressure load, normal to the stiffeners' surfaces, were modeled would the effect be noticable. However, the deformation of the inner shell was not expected to be closely simulated by the coupling techniques.

Actual construction of these regions required that the inner and outer surfaces be made with a number of small segments. A segment would occur at each primary rib and associated cooling channel. This scheme would assure the existence of common meshes. To accomodate this, a number of construction points were required. A numerical routine was developed to provide consistent placement of these points. After the wire frame model was built the surfaces and volumes were defined such that two surfaces existed at every rib location. From this, two continuous shells and a number of primary ribs were made. A further requirement was the use of an even number of elements in the housing's axial direction. This last restriction assured coincident nodes for the mid-plane stiffeners. Other than this restriction, the mesh configuration was optional, therefore, a simple uniform mesh was chosen. Due to the cylindrical shape of the housing, SUPERB's wavefront optimization routine performs very poorly. Thus, to optimize the elemental wavefront of the model, the element numbering must procede in a radial fashion.

The expectations for the linear thin shell model were limited. It was hoped that insight into the stressing mechanisms of the housing could be gained. Furthermore, familarization with the

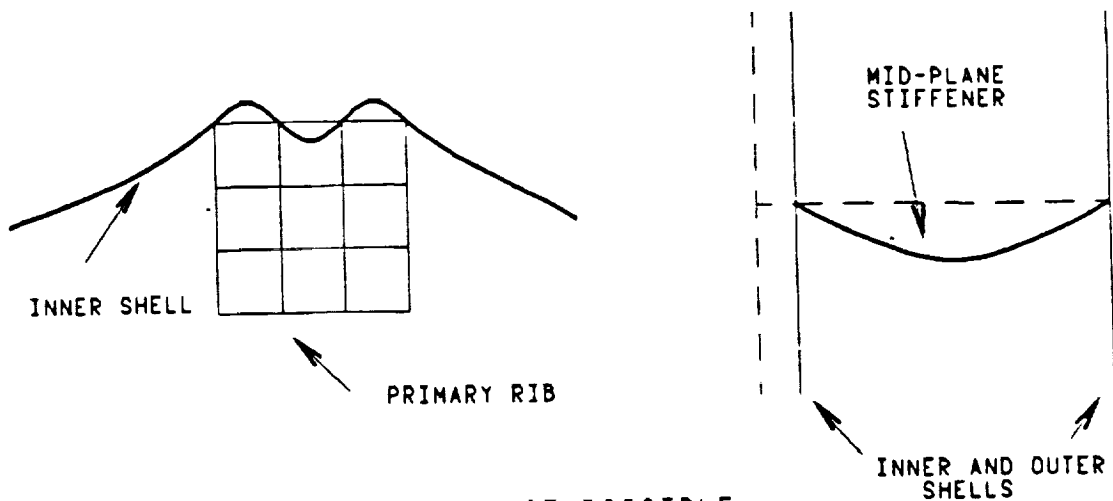


FIGURE 4.4: ILLUSTRATION OF POSSIBLE DEFORMATIONS BY UTILIZING THE COUPLING METHOD.

modeling system was anticipated. The compatibility problem, as well as the fact that linear thin shell elements could not model shearing actions, was expected to cause the stresses at locations X and N to be off the measured values. However, under longitudinal loading, location H was expected to model the experimental values quite closely since only a simple uniaxial load would be in play. Overall, the linear thin shell model was expected to provide an initial step in the construction sequence.

#### 4.11 ANALYSIS OF THE LINEAR THIN SHELL MODEL

After its completion, the linear thin shell model was subjected to the longitudinal tensile test described in chapter 3. As also mentioned in chapter 3, the loading distribution was analyzed. However, after analysis, it was determined that the actual loading distribution had little effect on the regions around strain gage rosettes. Thus, verification was completed by simply distributing the load across the bolt hole's length. Furthermore, various supporting methods were also applied. Basically, three types of supports were examined. Each one utilized the restraints described in chapter 3. But, as seen in figure 4.5, each supported different regions of the housing. For

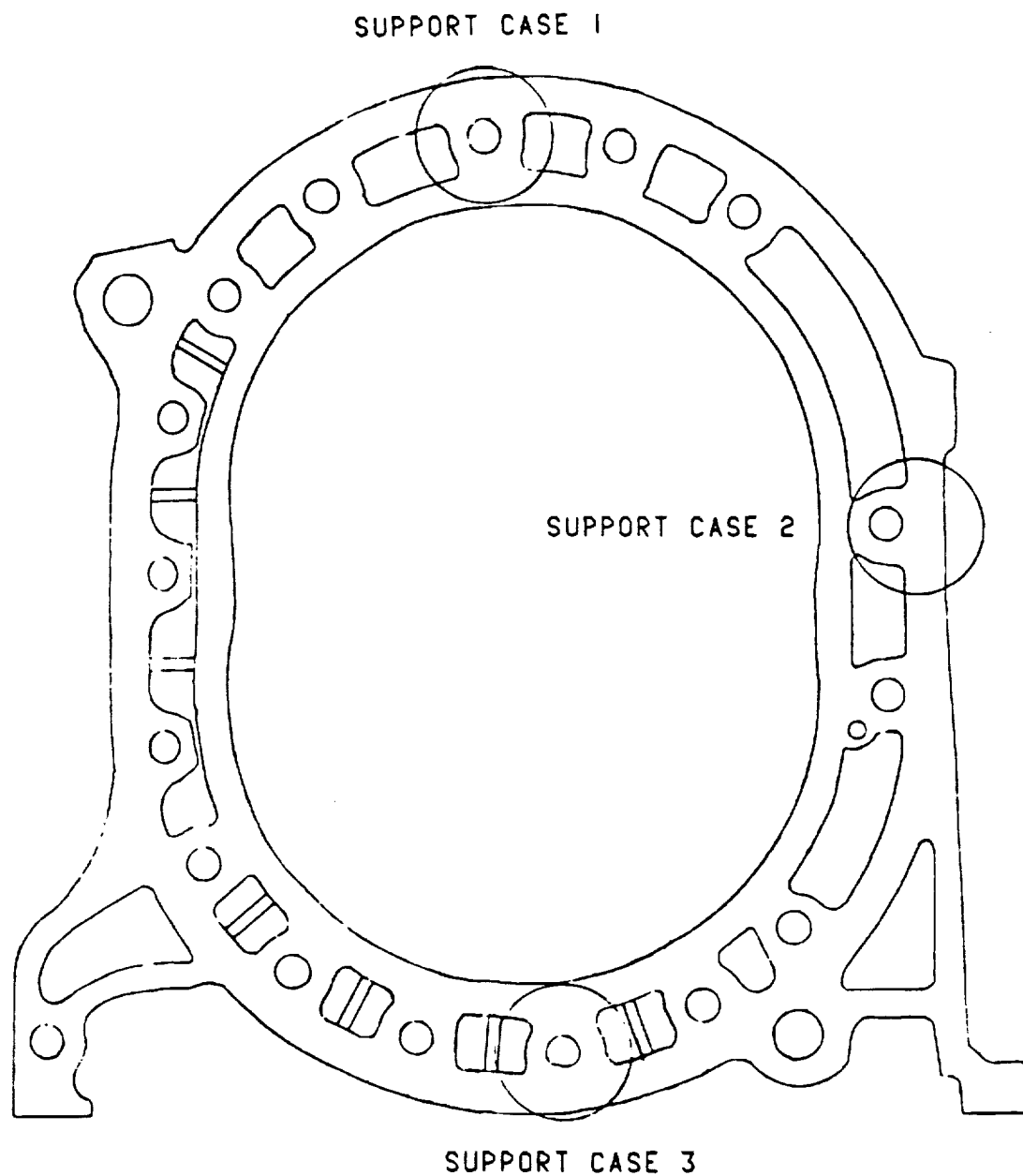


Figure 4.5: Various regions where support of the model was investigated.

simplicity, the groups will be referred to as support case 1, 2, and 3 as illustrated in figure 4.5. Results from this testing were obtained in the form of deformation plots and nodal stresses. For each support case, nodal stresses were compared to the experimental baseline data provided in table 1. Also, for each case, the particular loading and support of the model was confirmed through the use of the deformation plots (refer to figures 4.6, 4.7, and 4.8).

When testing was complete, the results were reviewed. From this, two conclusions were drawn. First, high stress gradients occurred around each rosette location, thus, comparison of stresses would need to be approximated. Second, of the three support cases, only case 2 enabled natural deformation of the housing to occur. The first of the conclusions is emphasized by figures 4.9-4.17. Even though only a few data points are available, the stress gradients are obvious. An illustration of the strain gage locations, shown in figure 4.18, will aid in the interpretation of the grid figures. The approximate rosette locations are shown, and the estimated stresses at each location are given in table 2. From table 2, the second conclusion can be drawn. Clearly, the rosette locations were affected by the nearby support of the housing. Consequently, support case 2 was chosen for the remaining testing. Finally, close inspection of the results indicated the accuracy of the linear thin shell model. Stresses, in each location, matched the measured values quite closely. Also, changes in the support cases were very easily predicted and, the existence of the stress gradients provided proof of the adequacy of the testing used to exercise the model.

#### 4.20 CONSTRUCTION OF THE PARABOLIC THIN SHELL MODEL

The second model in the construction sequence, the parabolic thin shell model, was basically chosen to analyze the mesh configuration. Since the same geometry could be utilized, the

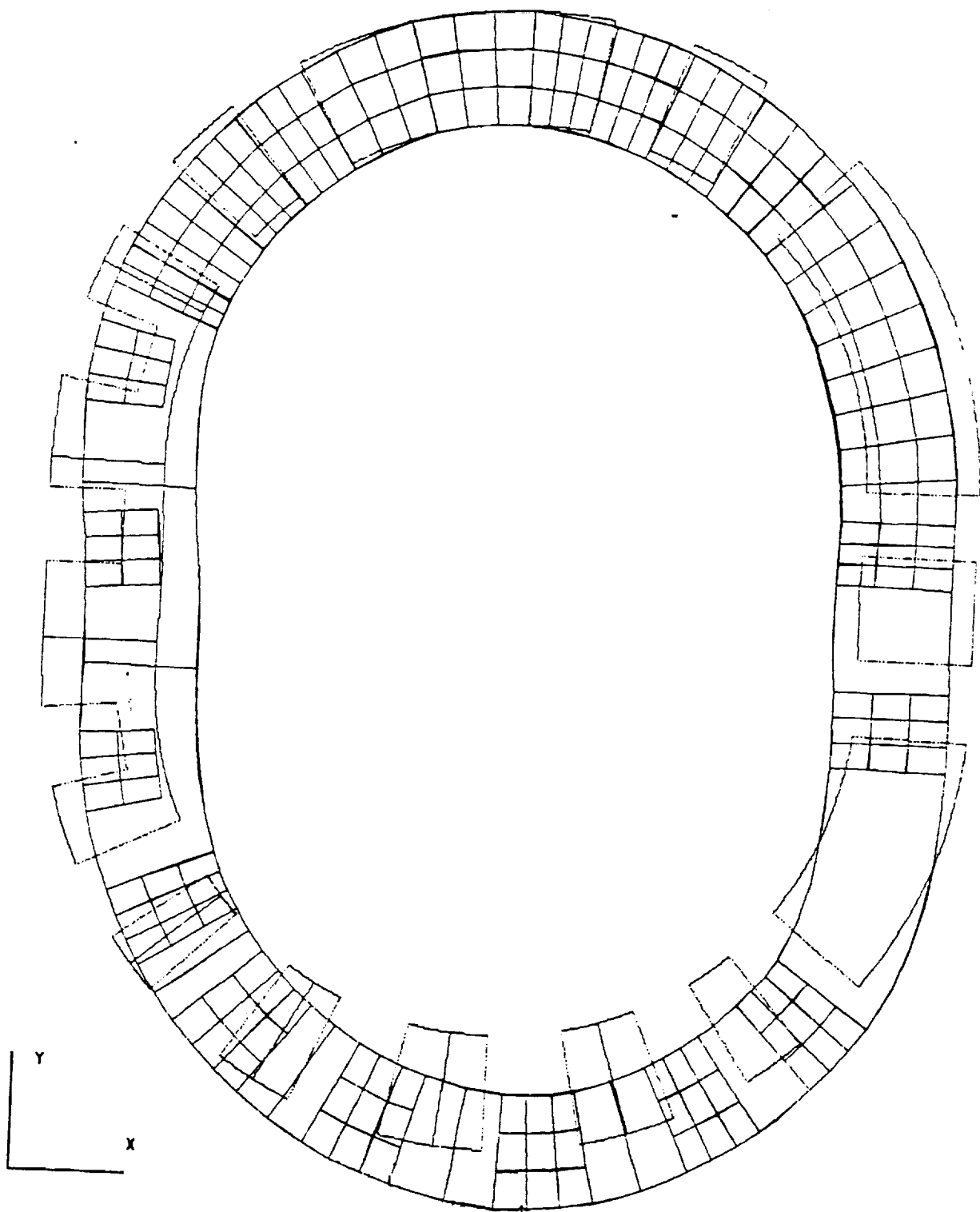


Figure 4.6: Deformation of linear thin shell model (support case 1).



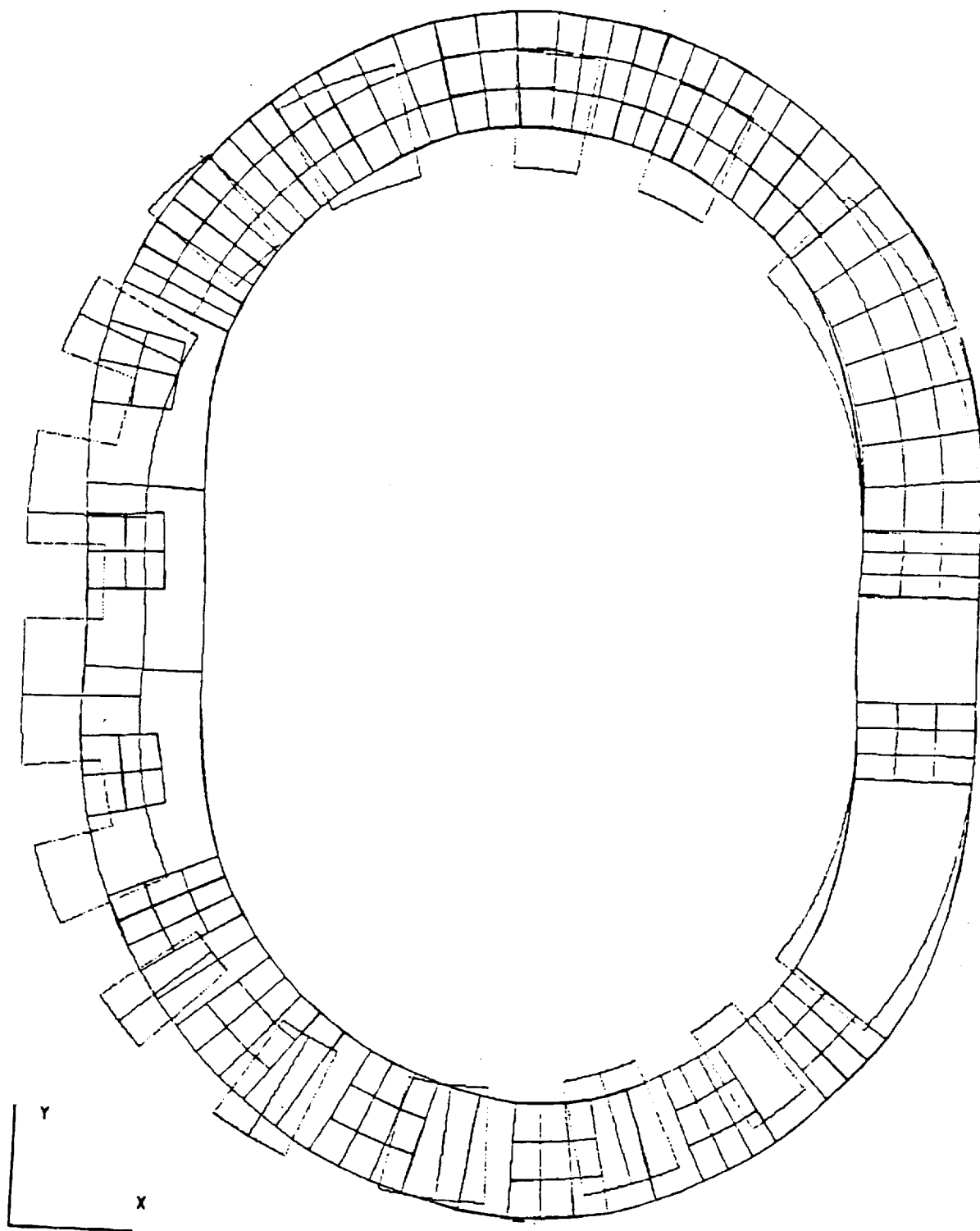


Figure 4.7: Deformation of linear thin shell model (support case 2).

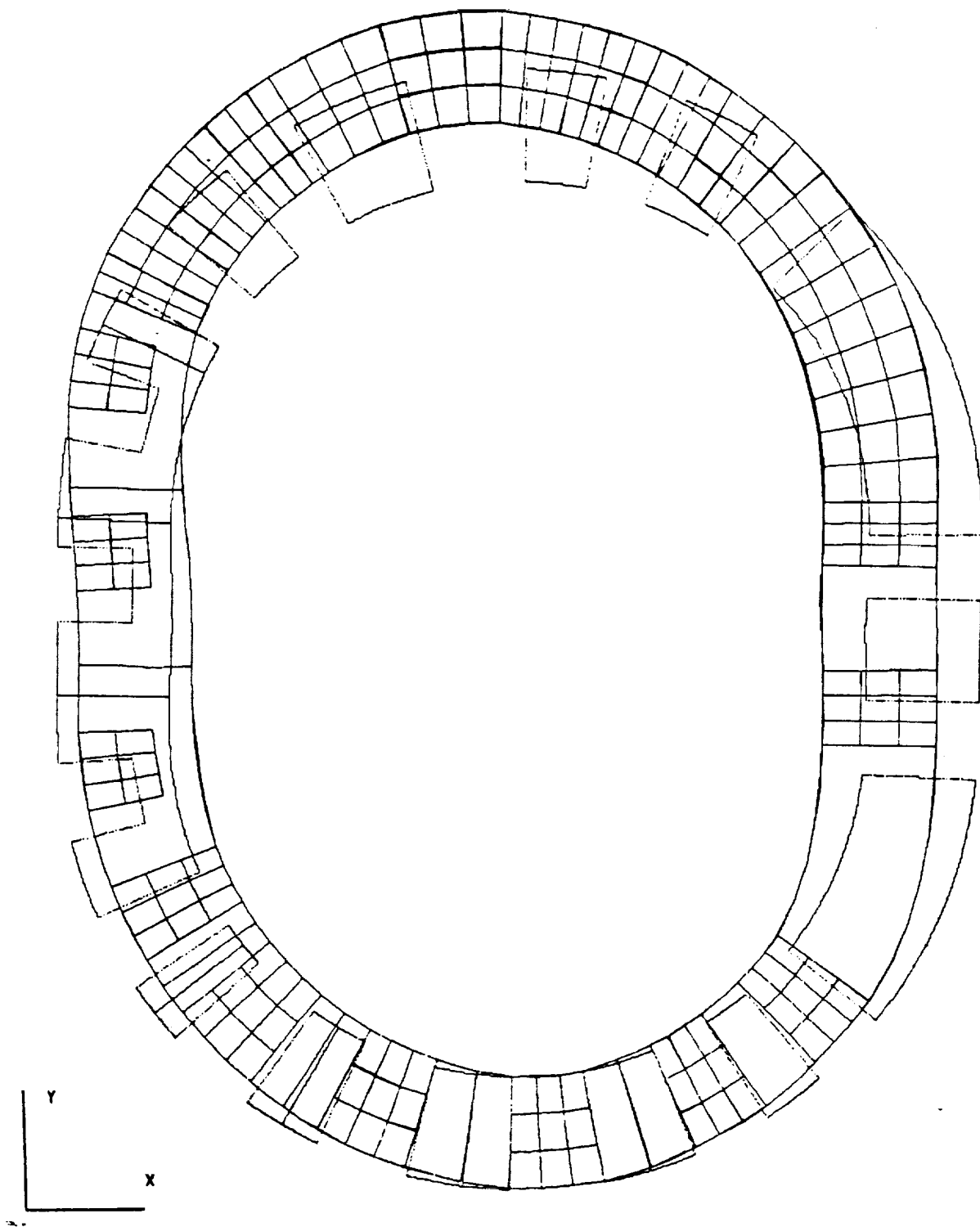
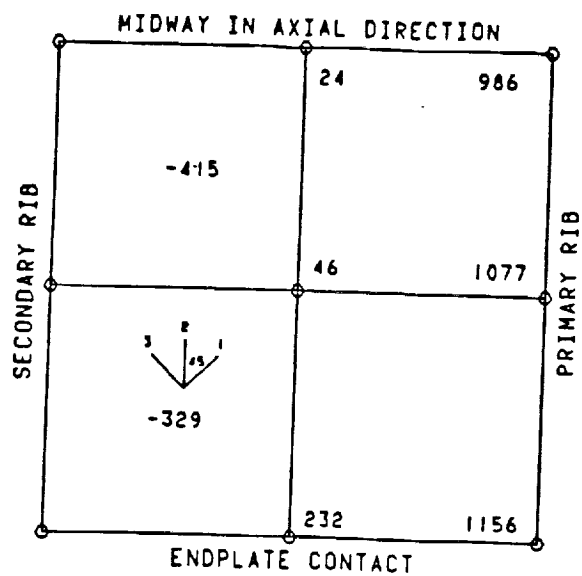
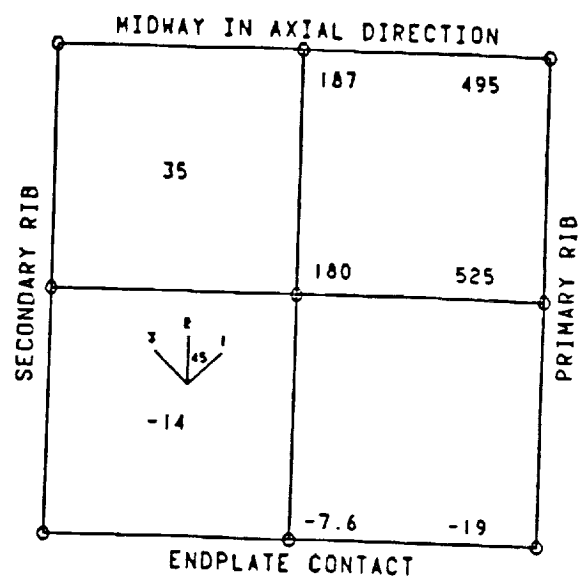


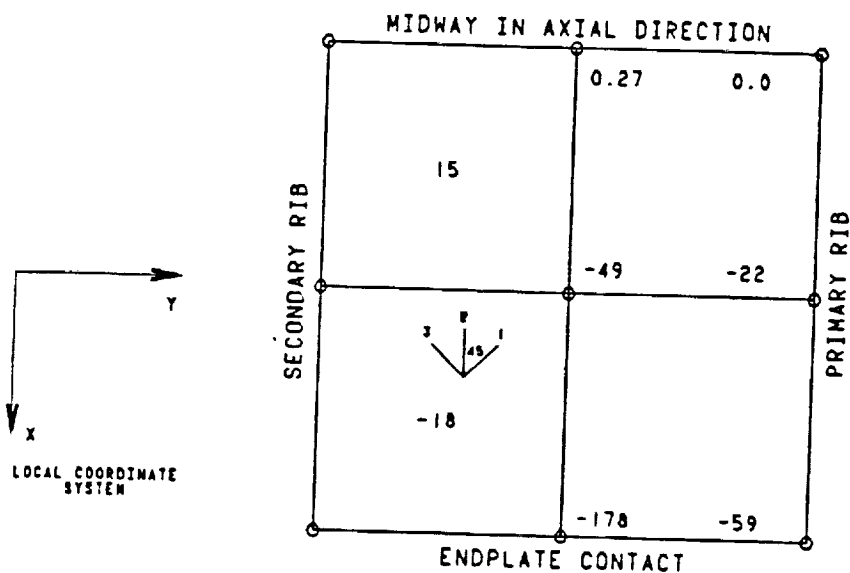
Figure 4.8: Deformation of linear thin shell model (support case 3).



STRESS Y (HOOP)



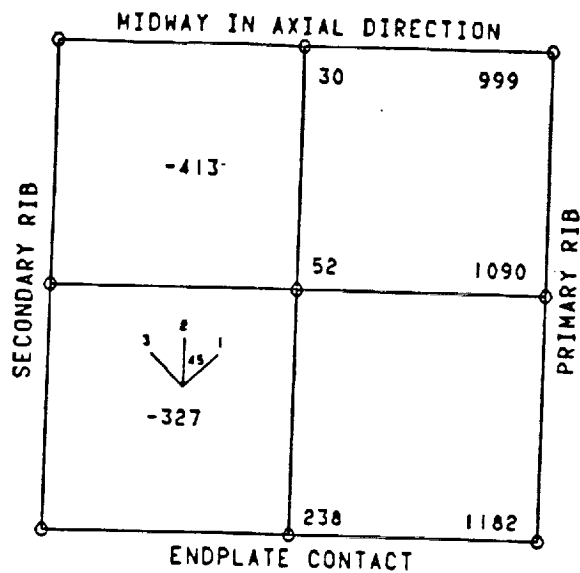
STRESS X



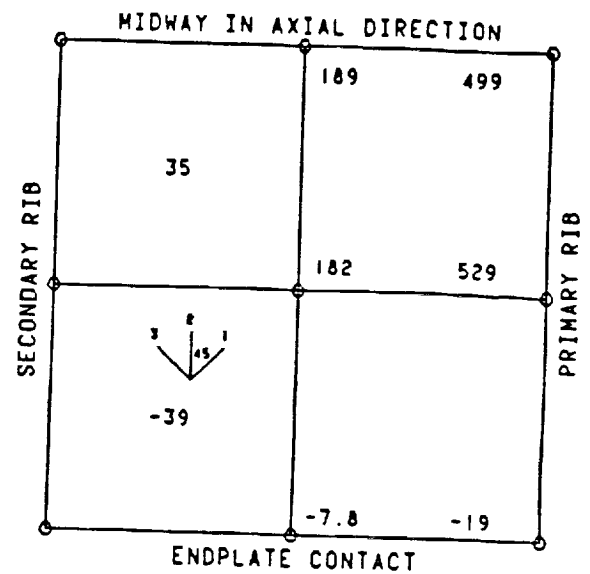
STRESS TAU-XY

NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

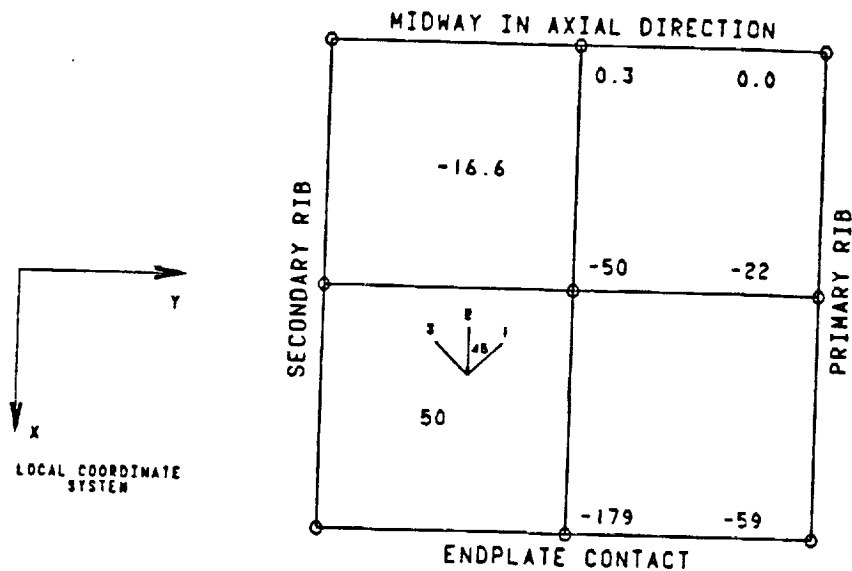
FIGURE 4.9: STRESSES AT LOCATION N  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE I



STRESS Y (HOOP)



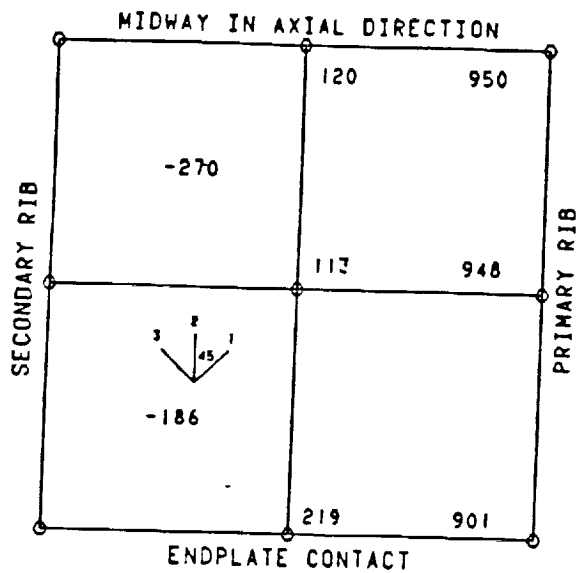
STRESS X



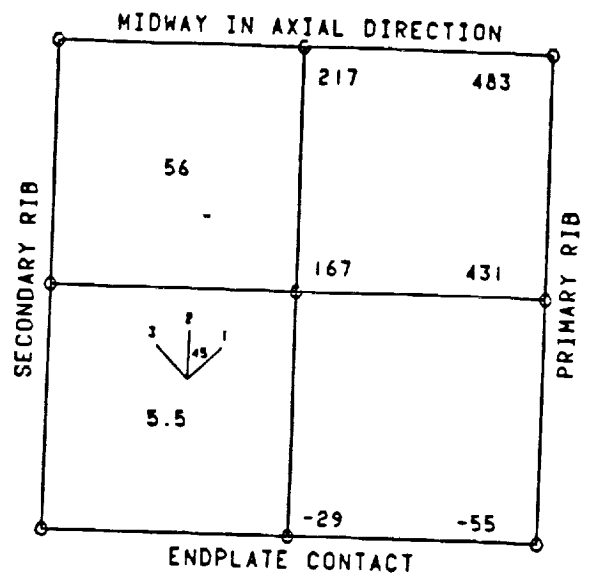
STRESS TAU-XY

NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

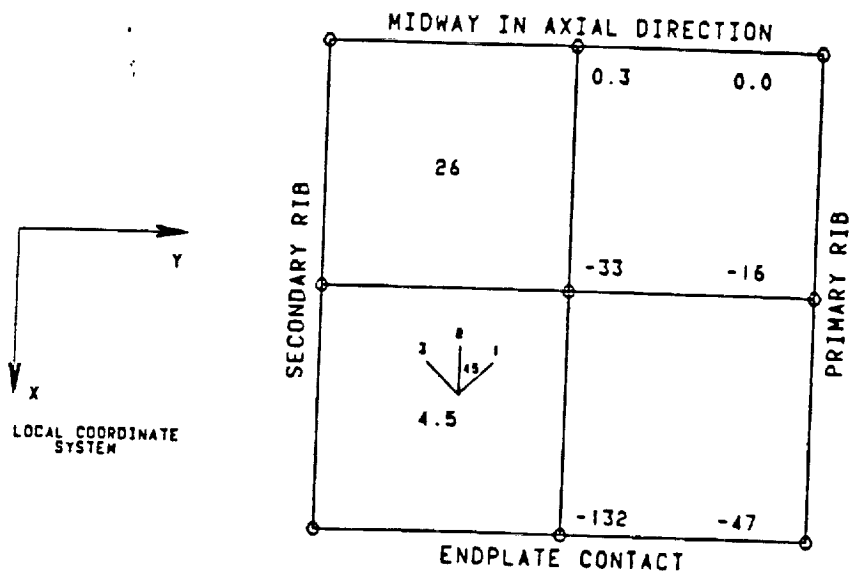
FIGURE 4.10: STRESSES AT LOCATION N  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE 2



STRESS Y (HOOP)



STRESS X



STRESS TAU-XY

NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 4.11: STRESSES AT LOCATION N  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE 3

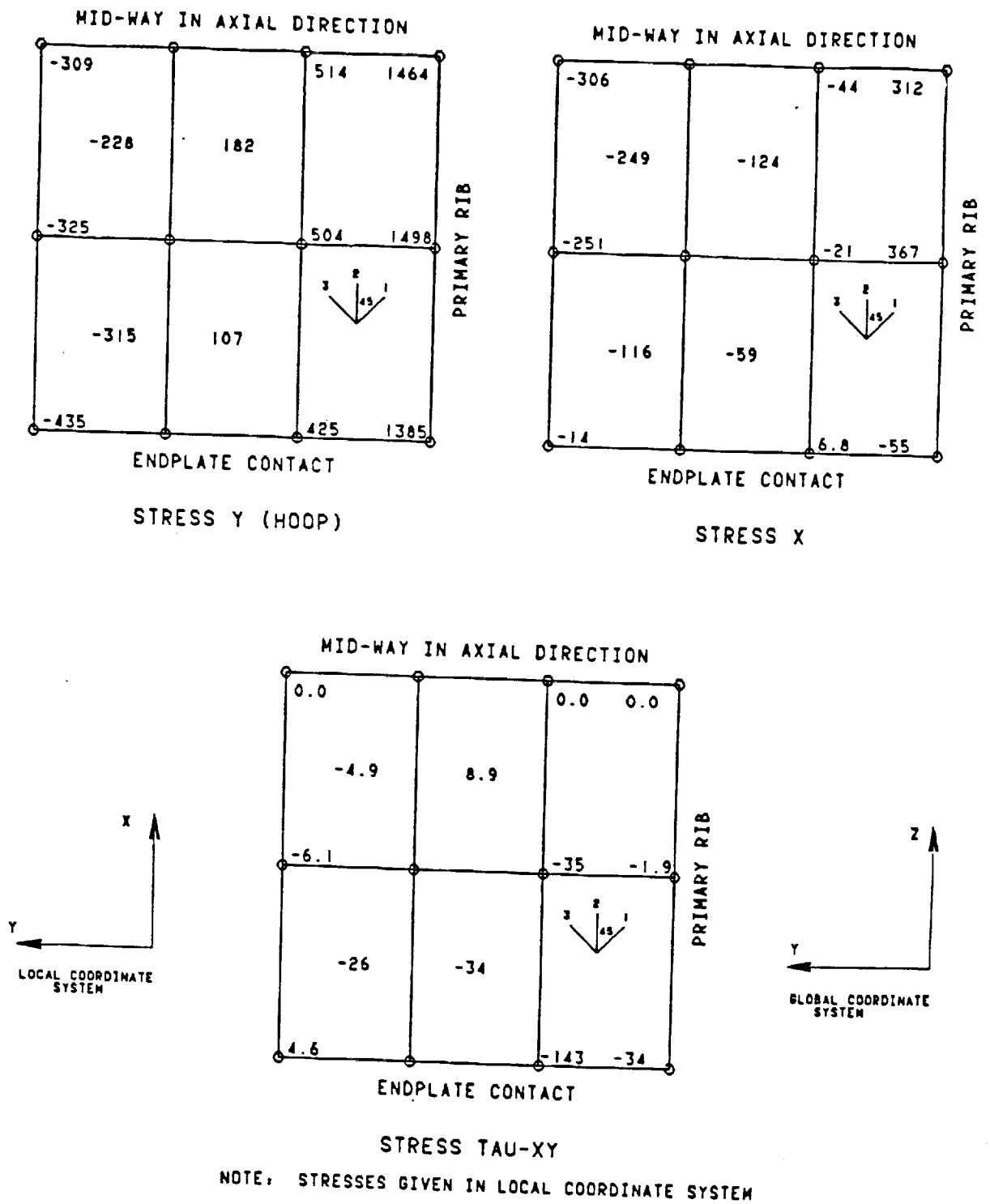
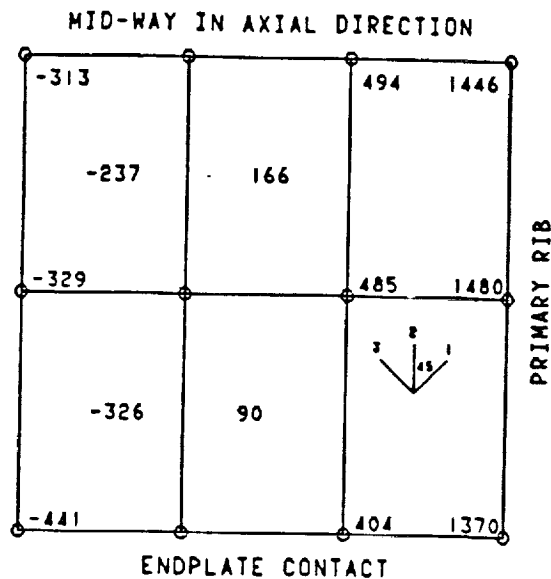
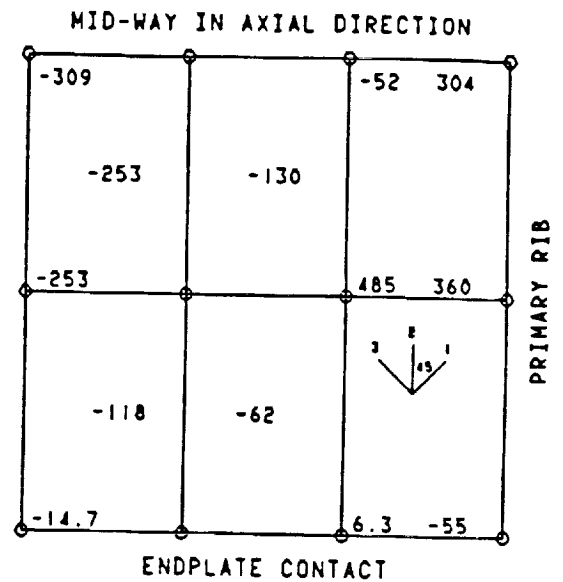


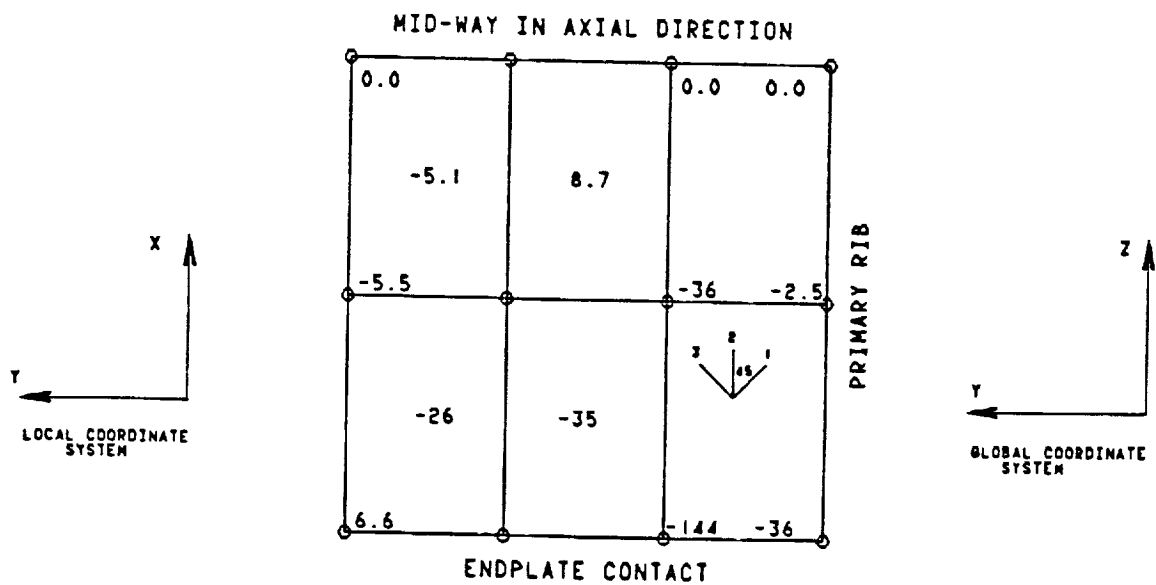
FIGURE 4.12: STRESSES AT LOCATION H  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE I



STRESS Y (HOOP)



STRESS X



STRESS TAU-XY

NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 4.13: STRESSES AT LOCATION H  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE 2

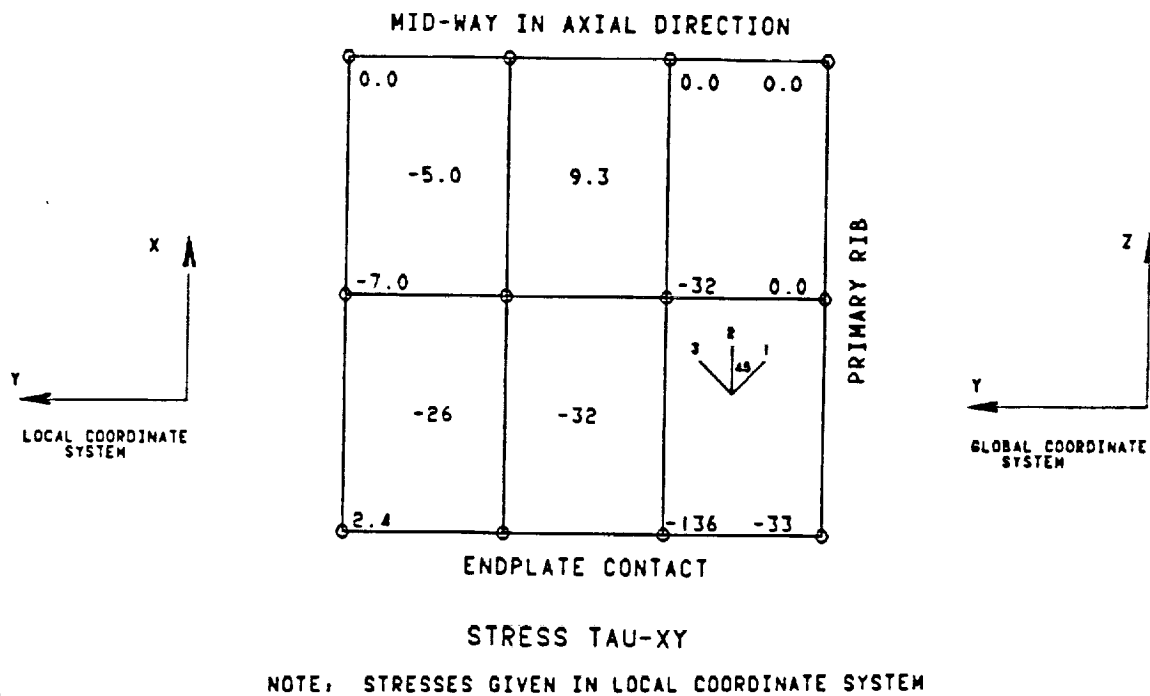
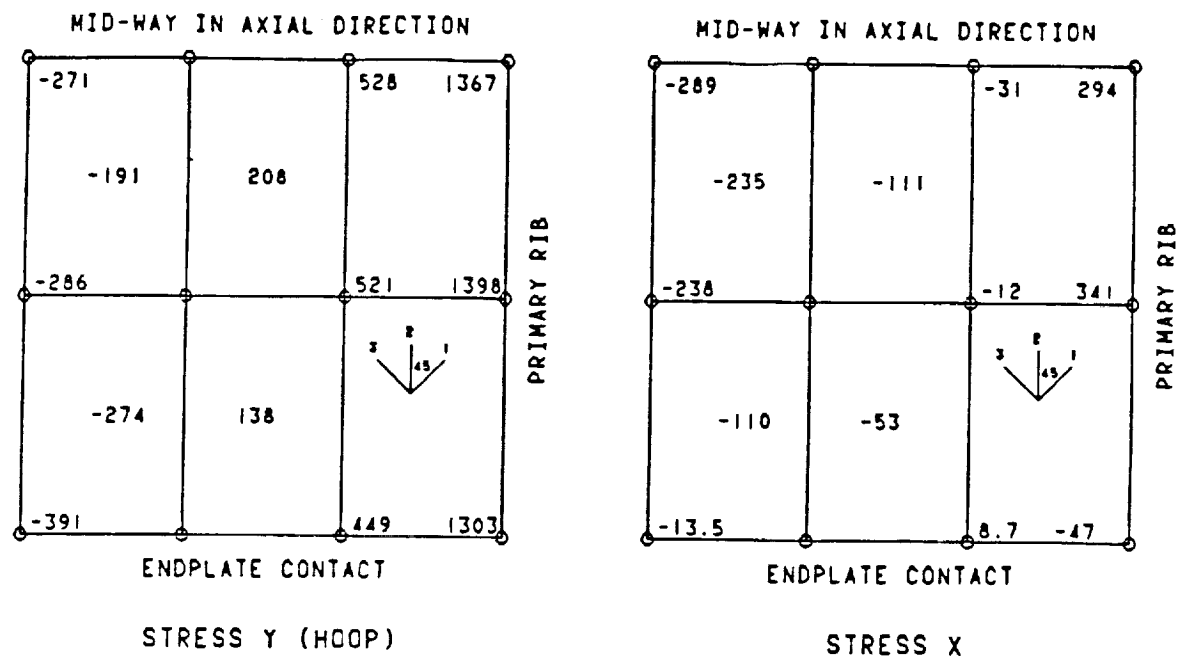


FIGURE 4.14: STRESSES AT LOCATION H  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE 3



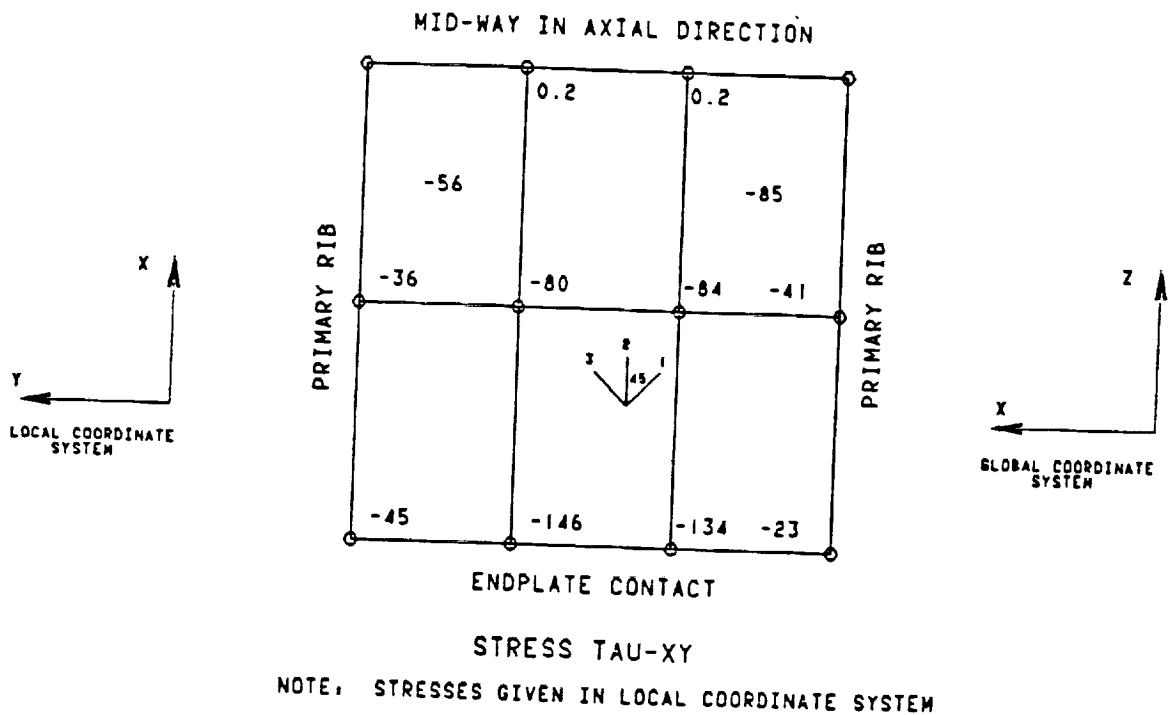
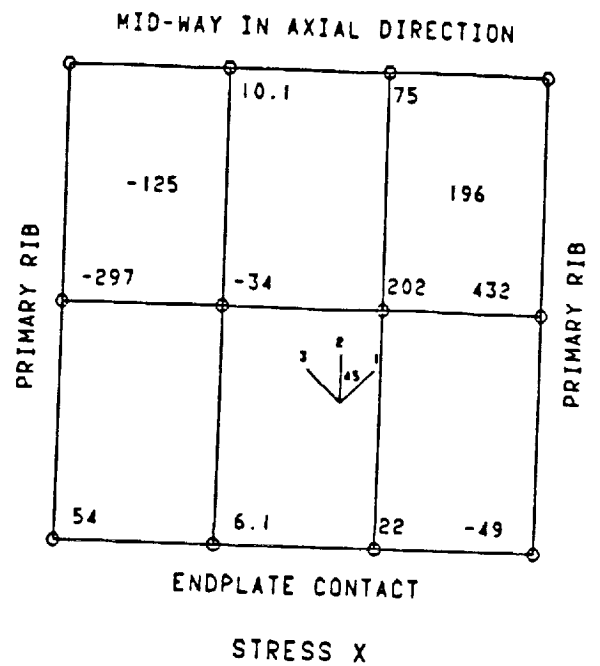
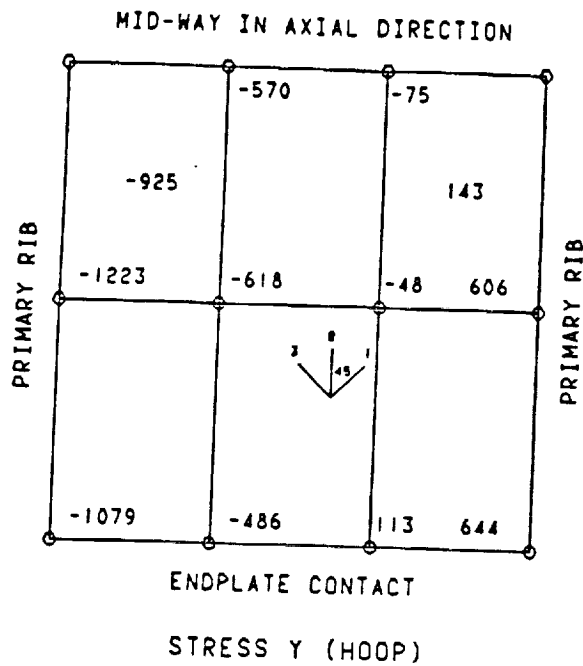


FIGURE 4.15: STRESSES AT LOCATION X  
LINEAR THIN SHELL MODEL  
RESTRAINT CASE 1

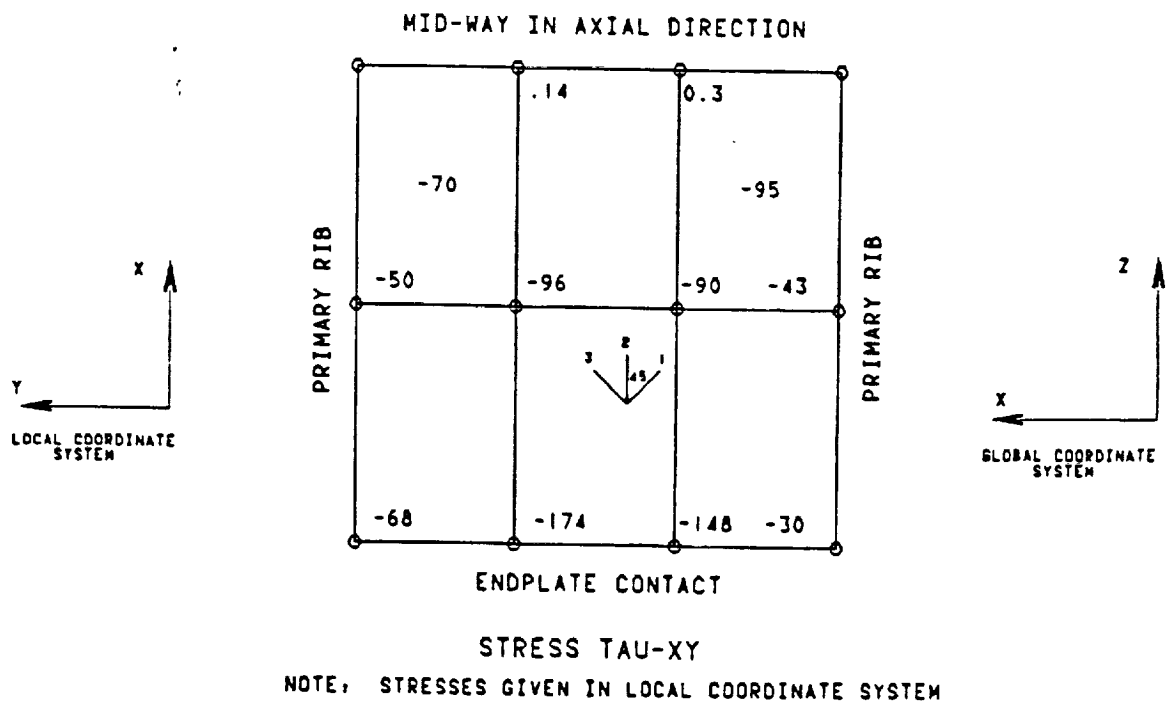
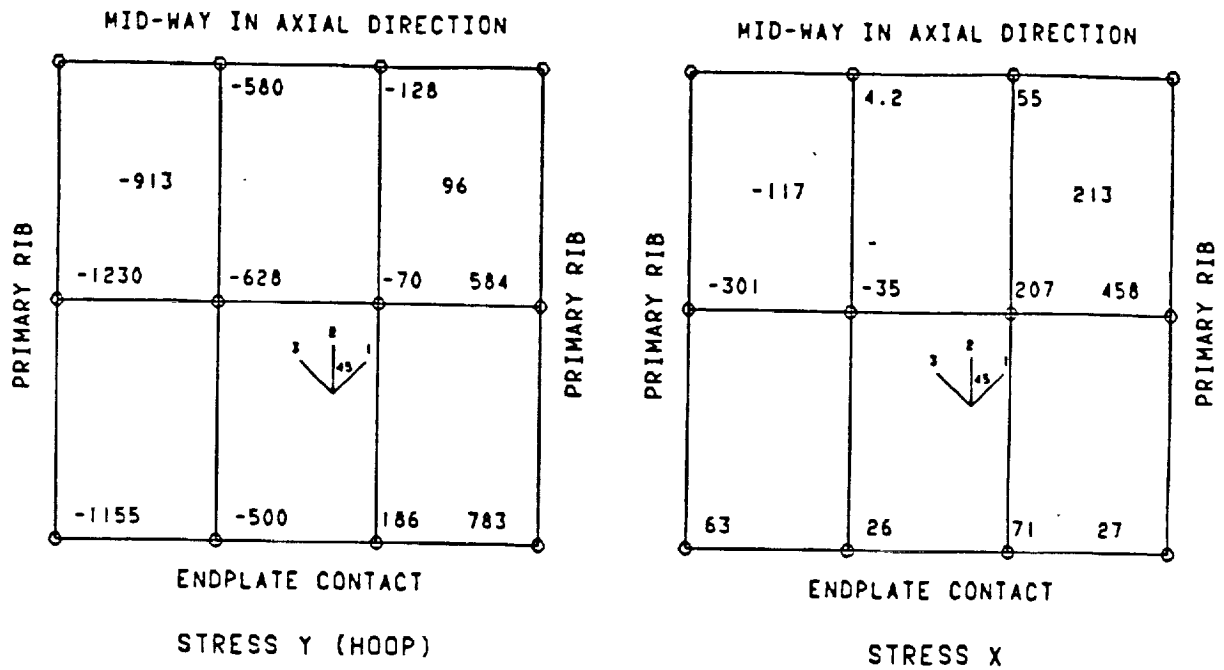


FIGURE 4.16: STRESSES AT LOCATION X  
 LINEAR THIN SHELL MODEL  
 RESTRAINT CASE 2

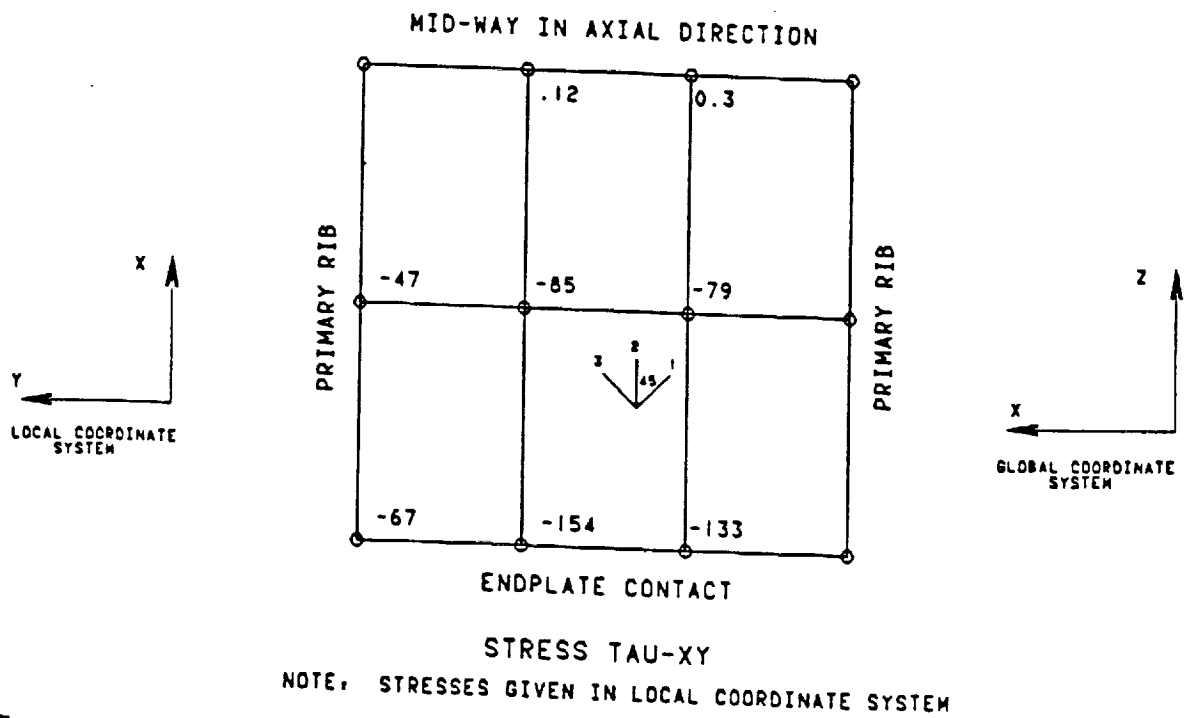
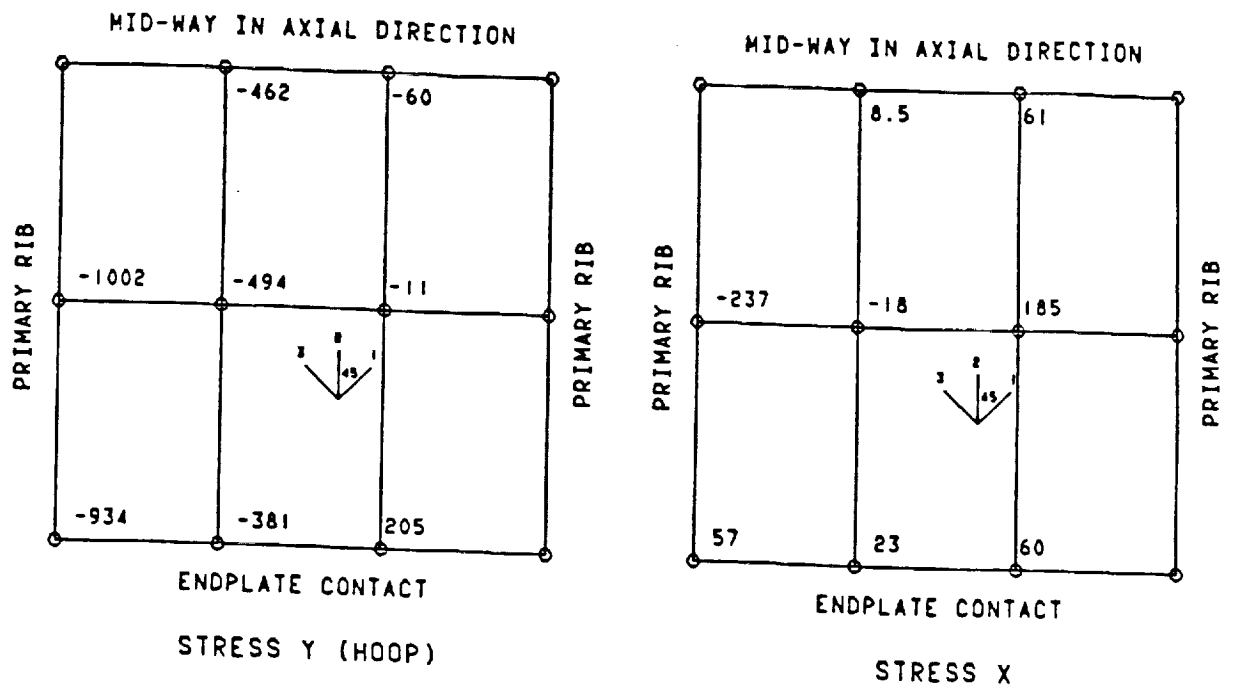


FIGURE 4.17: STRESSES AT LOCATION X  
 LINEAR THIN SHELL MODEL  
 RESTRAINT CASE 3

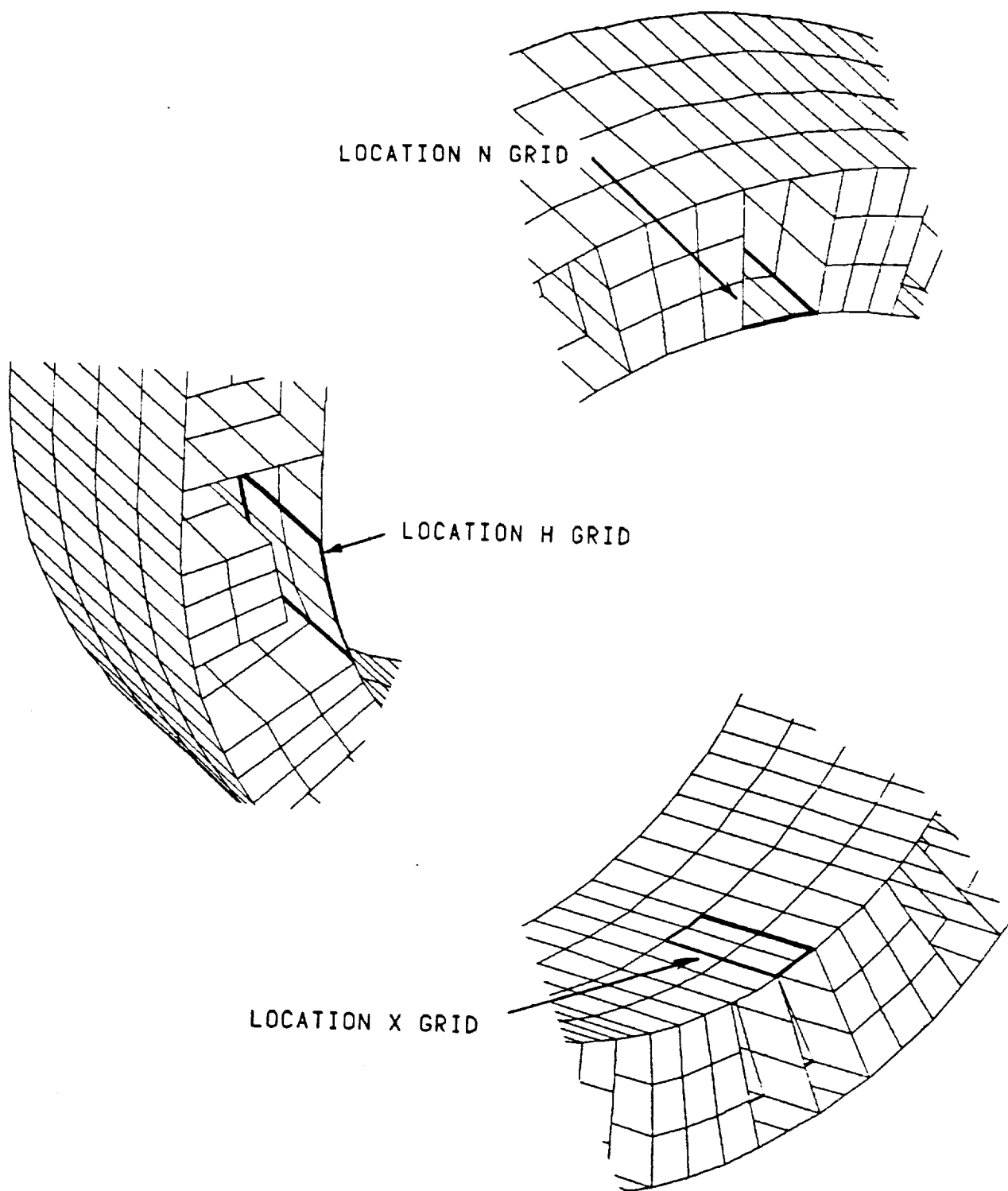


Figure 4.18: Illustration of strain gage rosette and stress grid locations.

LOCATION	ITEM	TIED * LOC. X	TIED * TRANS	TIED * LOC. N	EXP DATA
N	SIGMA-X	20	20	20	75
	SIGMA-Y	-350	-350	-200	-597
	TAU-XY	-15	30	0.0	-85
	SIGMA-P1	-4.3	22	20	86
	SIGMA-P2	-350	-352	-200	-608
	DIRECTION	2.5	-7.9	0.0	7.1
H	SIGMA-X	100	100	100	-9.3
	SIGMA-Y	700	700	700	702
	TAU-XY	-40	-40	-40	36
	SIGMA-P1	702	702	702	704
	SIGMA-P2	97	97	97	-11
	DIRECTION	-3.4	-3.4	-3.4	2.9
X	SIGMA-X	120	120	120	85
	SIGMA-Y	40	100	60	101
	TAU-XY	-100	-100	-100	-77
	SIGMA-P1	187	210	194	170
	SIGMA-P2	-27	9.5	-14	15
	DIRECTION	34	42	37	47

\*NOTE: POSITIVE DIRECTIONS INDICATE CLOCKWISE ROTATION OF THE AXES.

### STRESSES GIVEN IN PSI

Table 2: Results from linear thin shell model.

effect of modifications in the mesh could easily be investigated. However, the performance of the linear thin shell model indicated that the need for mesh refinements was limited. Therefore, the major drive behind the construction of the parabolic thin shell model was verification of the existing mesh. The introduction of parabolic elements provided the capability to model shearing actions as well as additional data points. Both of these features were expected to produce better agreement with the measured values. Furthermore, although stresses were expected to be slightly higher than those obtained previously, similar results were anticipated.

#### 4.21 ANALYSIS OF THE PARABOLIC THIN SHELL MODEL

With the use of the same geometry, construction of the parabolic thin shell model was easily completed. When finished, the model was similarly subjected to the longitudinal tensile test. Support of the housing was achieved by utilizing support

case 2. In addition, results were obtained in the same manner as before. As in the linear model, high stress gradients occurred around each rosette location. The results resembled the gradients obtained with the linear model. These results are shown in figures 4.20-4.22, and the approximated stresses are listed in table 3. Also, loading and support of the model was confirmed through the use of the deformation plot shown in figure 4.19.

The concurrence of the parabolic thin shell model's results to the results from the linear thin shell model proves convergence of the mesh used. Furthermore, the expected slight increase in the stresses obtained, furnishes added confidence in the model's accuracy. Therefore, at this point in the study, it was felt that the structural characteristics of the housing were modeled quite adequately.

LOCATION	ITEM	MODEL RESULTS	EXP. DATA
N	SIGMA-X	50	75
	SIGMA-Y	-300	-597
	TAU-XY	-55	-85
	SIGMA-P1	58	86
	SIGMA-P2	-308	-608
	DIRECTION	8.7	7.1
H	SIGMA-X	60	-9.3
	SIGMA-Y	700	702
	TAU-XY	-40	36
	SIGMA-P1	702	704
	SIGMA-P2	57	-11
	DIRECTION	-3.6	2.9
X	SIGMA-X	80	85
	SIGMA-Y	100	100
	TAU-XY	-75	-76
	SIGMA-P1	166	170
	SIGMA-P2	14	15
	DIRECTION	48	48

NOTE: POSITIVE DIRECTION INDICATES  
CLOCKWISE ROTATION OF AXES.

STRESSES GIVEN IN PSI

Table 3: Results obtained from parabolic thin shell model.

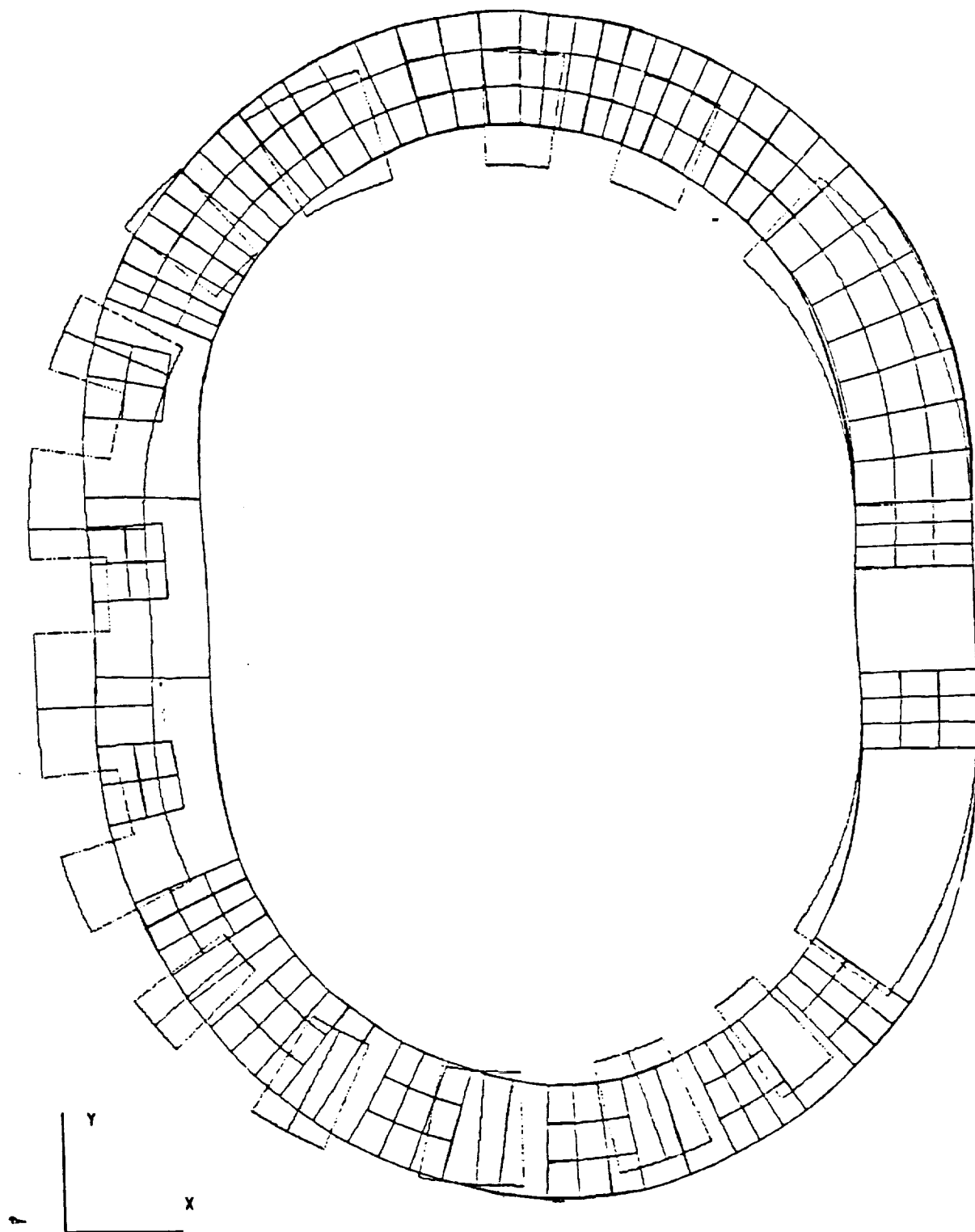


Figure 4.19: Deformation of the parabolic thin shell model subjected to a 1000 lb. tensile load

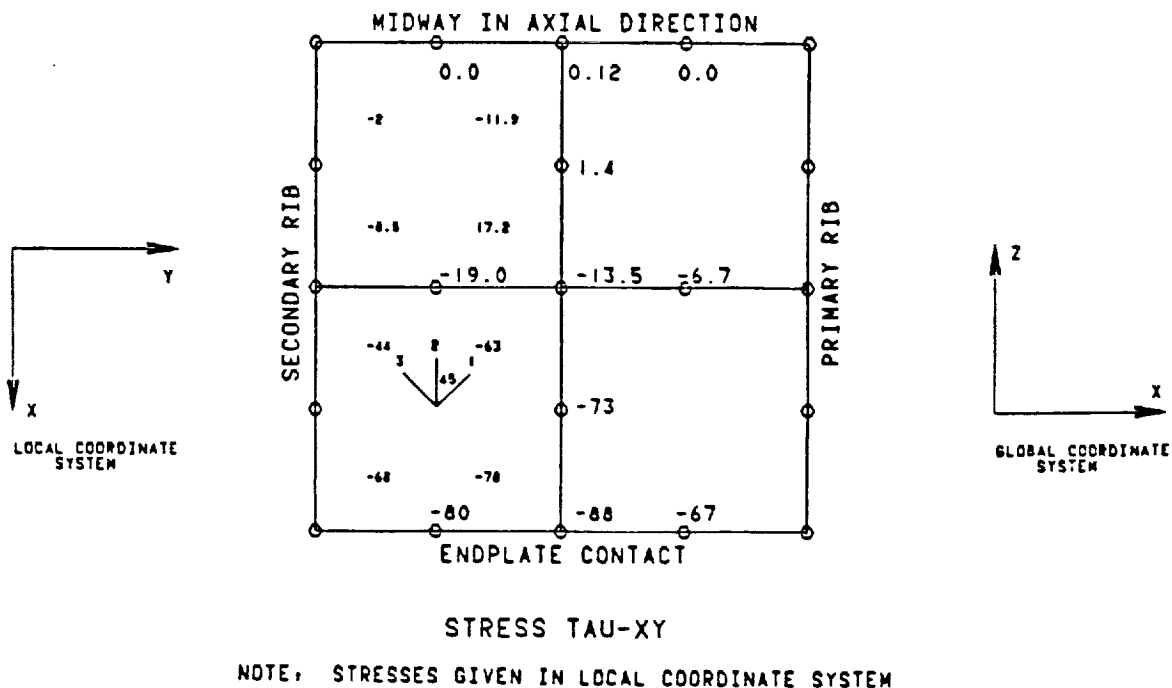
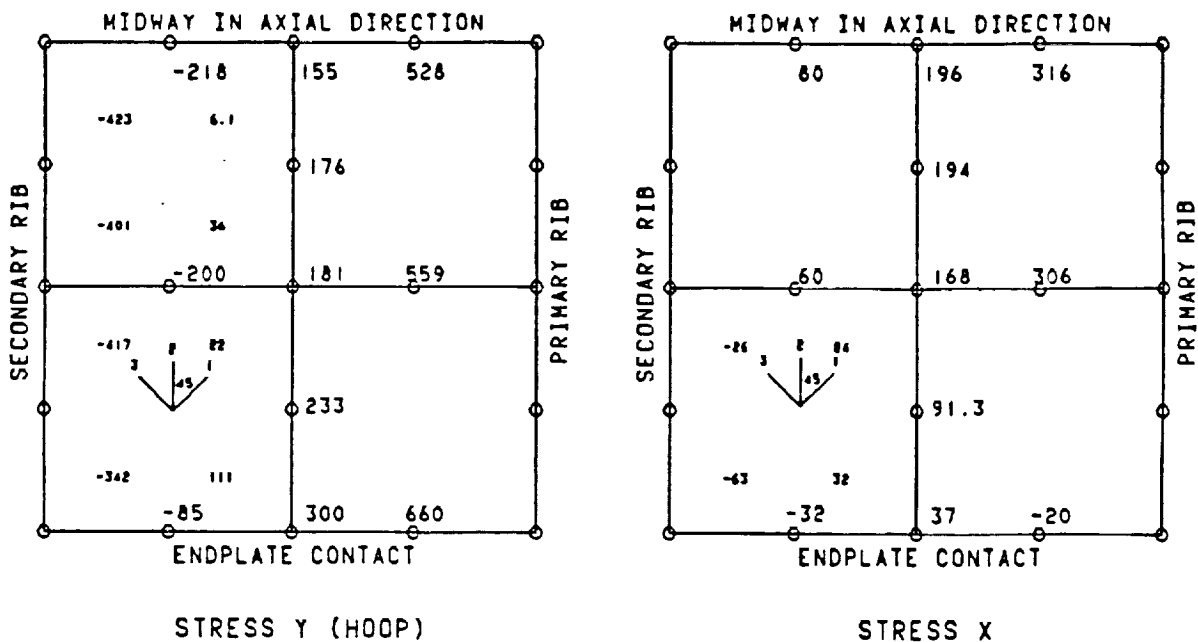
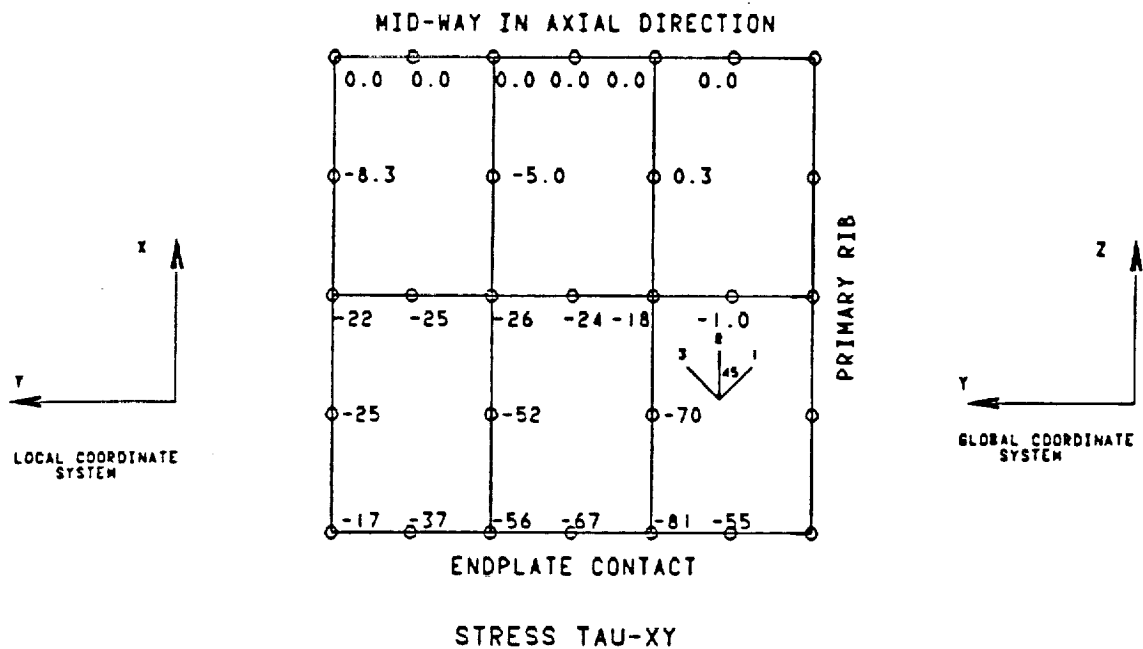
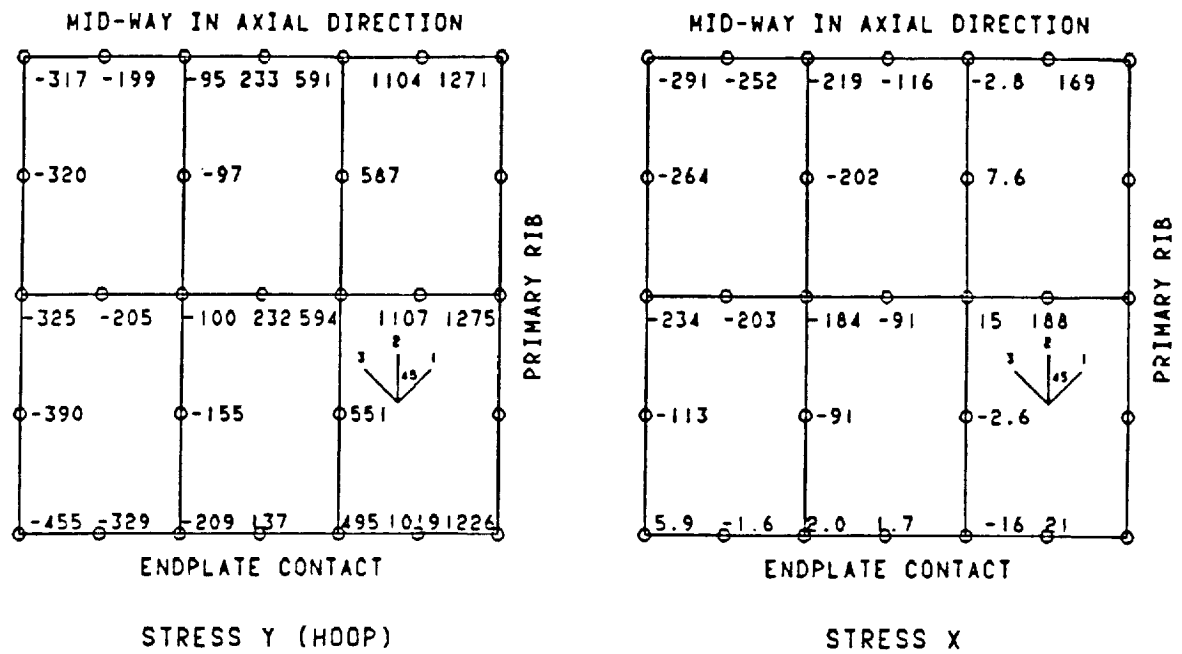


FIGURE 4.20: STRESSES AT LOCATION N  
PARABOLIC THIN SHELL MODEL  
RESTRAINT CASE 2





NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 4.21: STRESSES AT LOCATION H  
PARABOLIC THIN SHELL MODEL  
RESTRAINT CASE 2

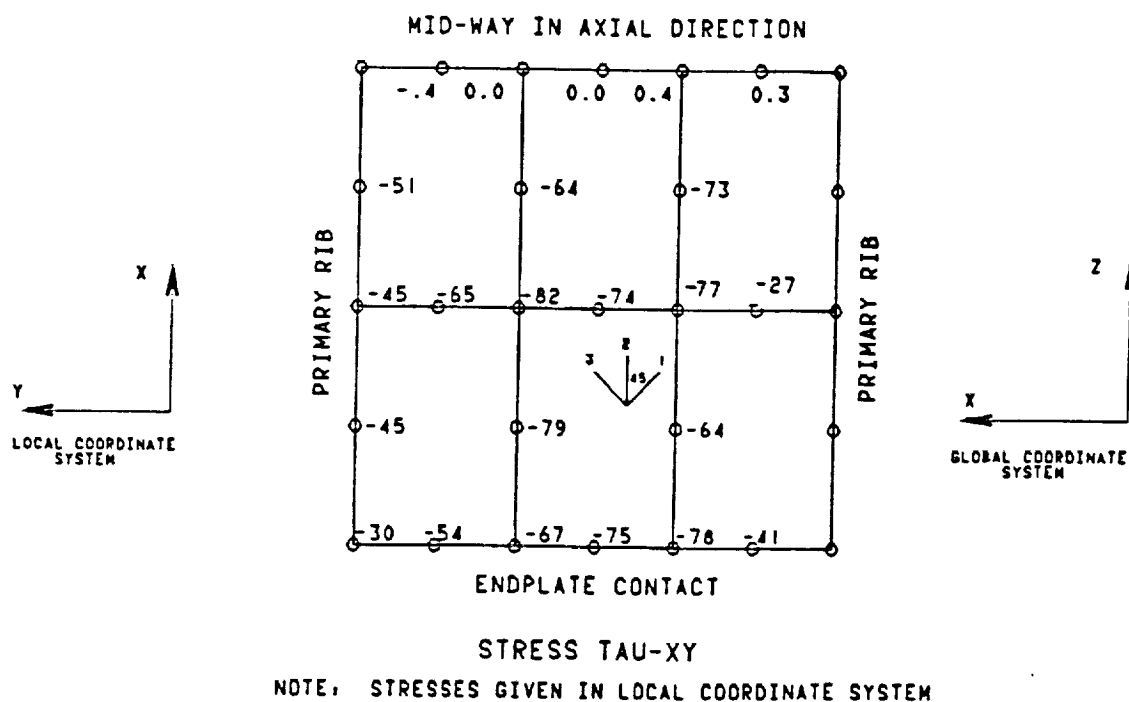
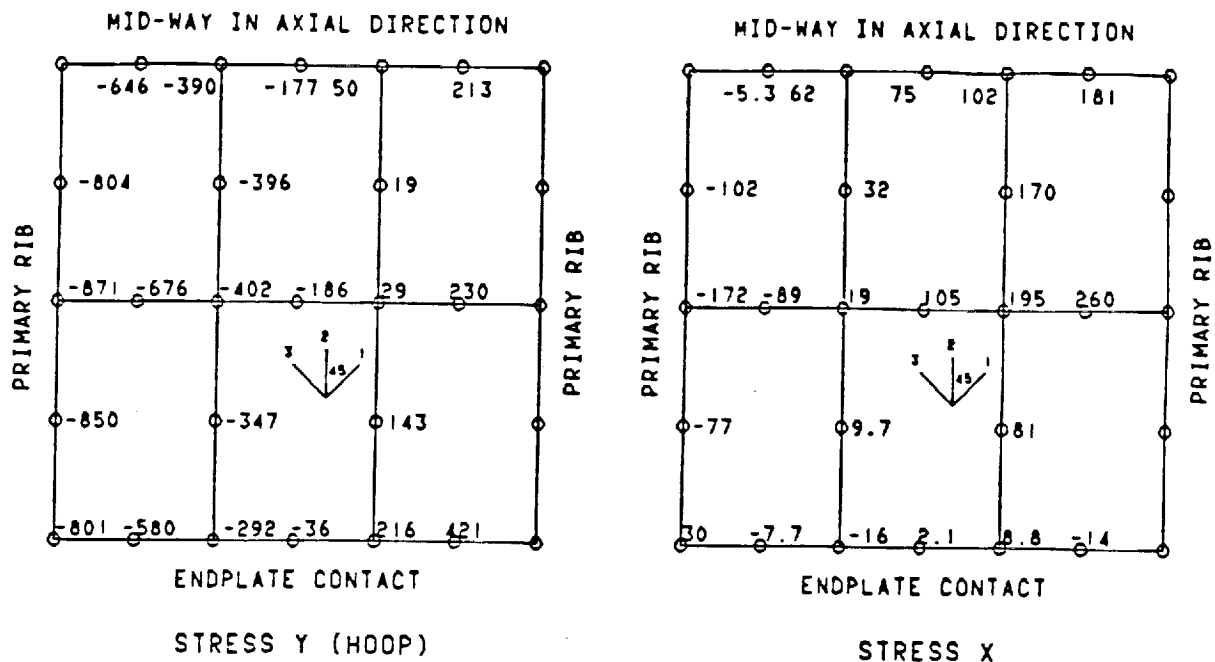


FIGURE 4.22: STRESSES AT LOCATION X  
PARABOLIC THIN SHELL MODEL  
RESTRAINT CASE 2

#### 4.30 CONSTRUCTION OF THE PARABOLIC THICK SHELL MODEL

Once the adequacy of the modeling techniques used was confirmed, construction of a model that could support a thermal analysis was needed. The third model, the parabolic thick shell model, encompassed this requirement. With the introduction of the parabolic thick shell model, shown in figure 4.23, the configuration of the FEM of the housing approached the form that was introduced in chapter 2. In this model, SUPERB's thick shell elements replaced the thin shell elements previously used to model the inner and outer shells. The remaining geometry, however, continued to utilize the same element types. Also, the same mesh configurations were employed.

Construction of the parabolic thick shell model was noticeably more complicated than the thin shell models, due to the fact that 3-D representations of the inner and outer shells were required. Once more, the spark plugs, engine mounts, and both ports were not modeled. The mesh size remained the same and construction methods were unchanged. However, one major problem surfaced after analysis had begun. Although the compatibility problems that occurred in the thin shell models were solved by using thick shell elements, the use of these elements produced the same problems in the mid-plane stiffener and secondary rib region of the thick shell models. Since nodal rotations in these regions were no longer supported as they were in the thin shell models, deformations, as seen in figure 4.24, could occur. The solution to this problem was quite innovative. A beam element which resisted bending but offered no resistance to uniaxial loading was introduced. By "burying" the beams into the thick shell and solid elements, as shown in figure 4.25, the stiffness of the regions was greatly increased.

It was anticipated that the outcome of the parabolic thick shell model would concur with the previous thin shell models. Since the structural qualities of the thick shell elements

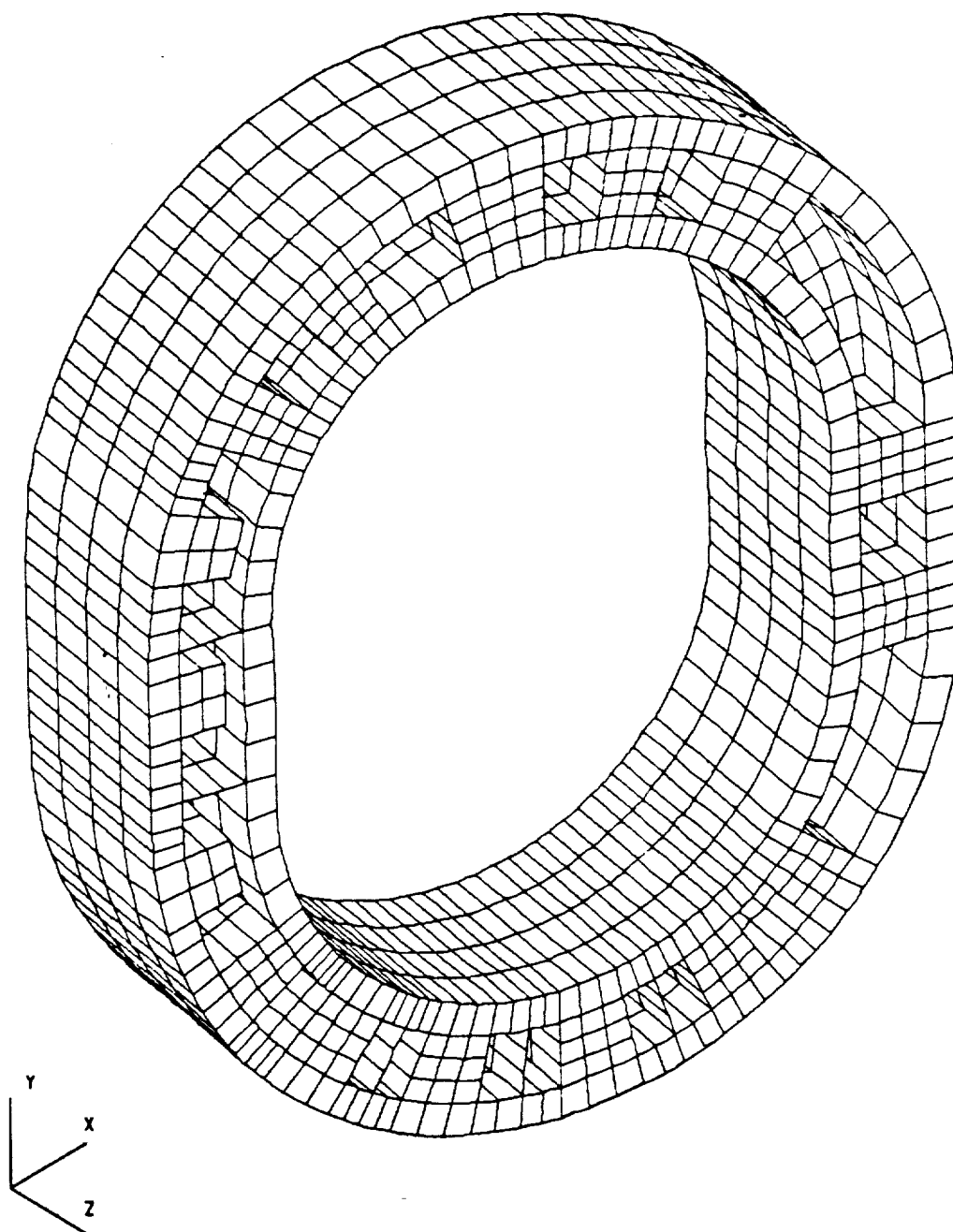


Figure 4.23: Illustration of the parabolic thick shell model.

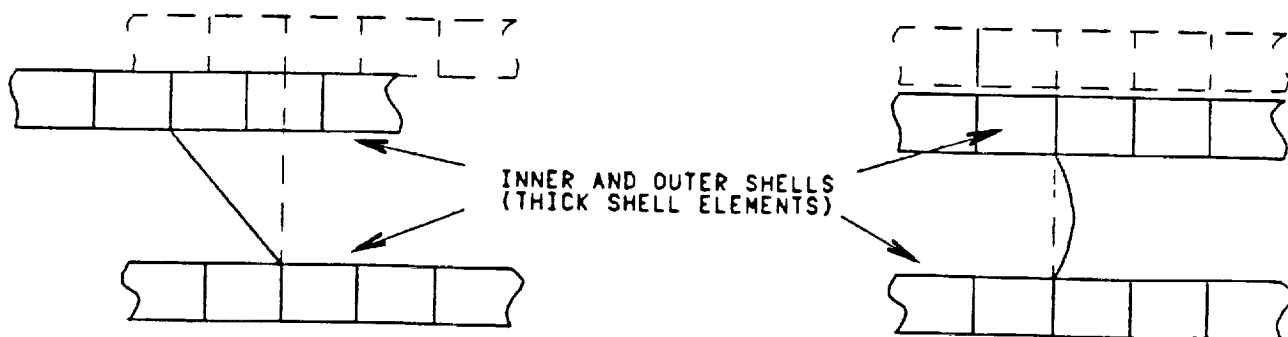


FIGURE 4.24: ILLUSTRATION OF POSSIBLE DEFORMATIONS OF HOUSING WITHOUT ELEMENTAL ROTATIONS SUPPORTED.

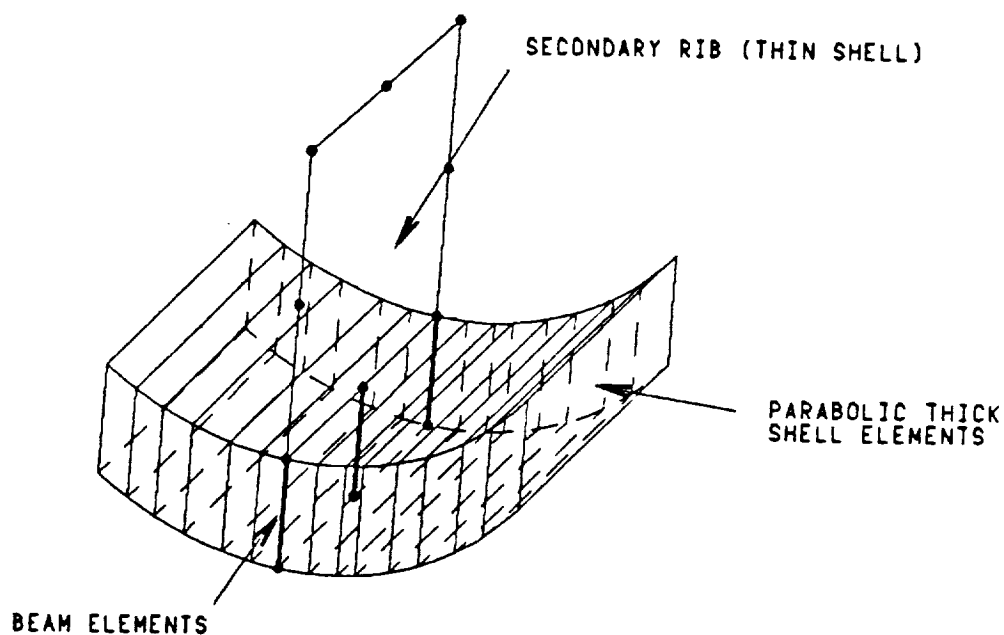


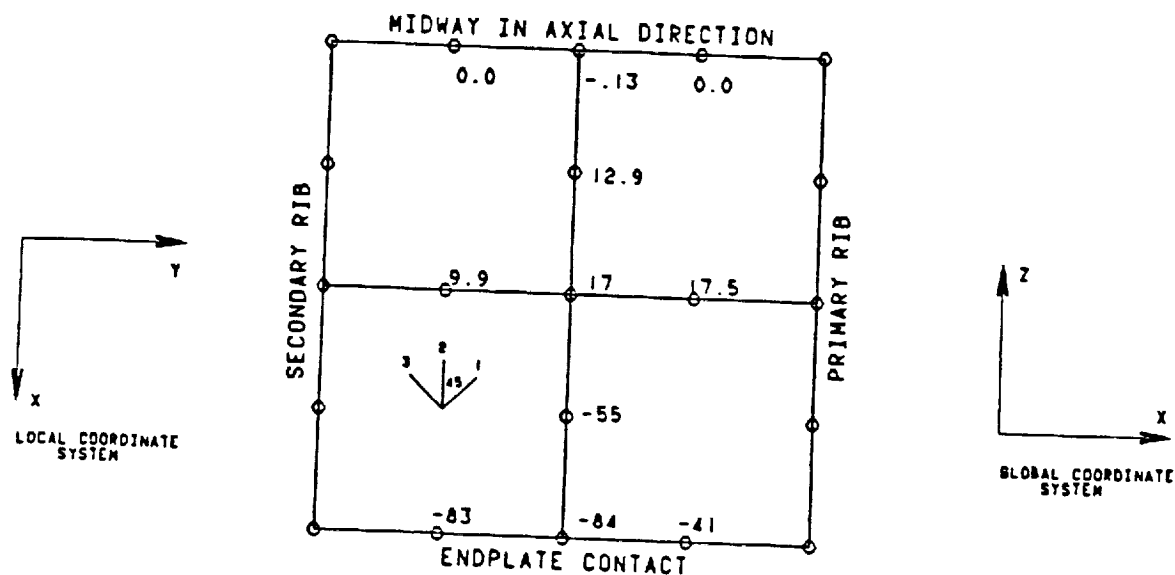
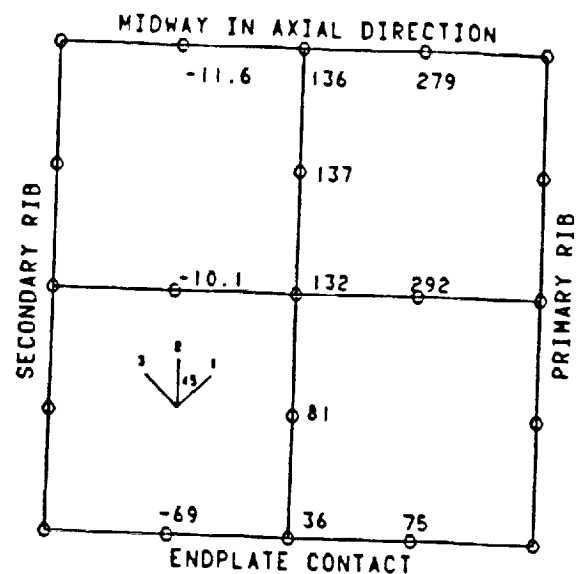
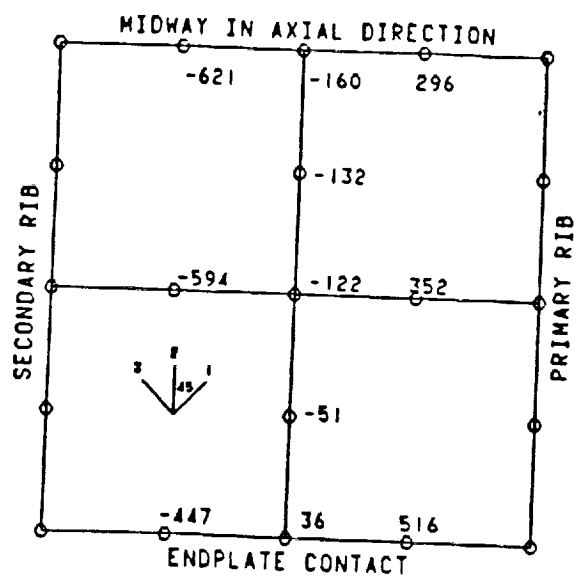
FIGURE 4.25: ILLUSTRATION OF THE INTRODUCTION OF BEAM ELEMENTS.

closely matched those of the thin shell elements, the stresses were not expected to be affected. The elimination of the compatibility problems around the primary ribs, however, would tend to make the model predict the deformation of the inner shell more closely. Also, it was predicted that the introduction of beam elements would not only provide adequate support in the regions involved, but also their effect on the inner and outer shells was expected to be negligible. This later expectation was due to the fact that the thick shell elements should "classically" deform, planes would remain planes. In conclusion, it was felt that the parabolic thick shell model would introduce thermal modeling capability but not effect the excellent structural modeling capabilities already shown.

#### 4.31 ANALYSIS OF THE PARABOLIC THICK SHELL MODEL

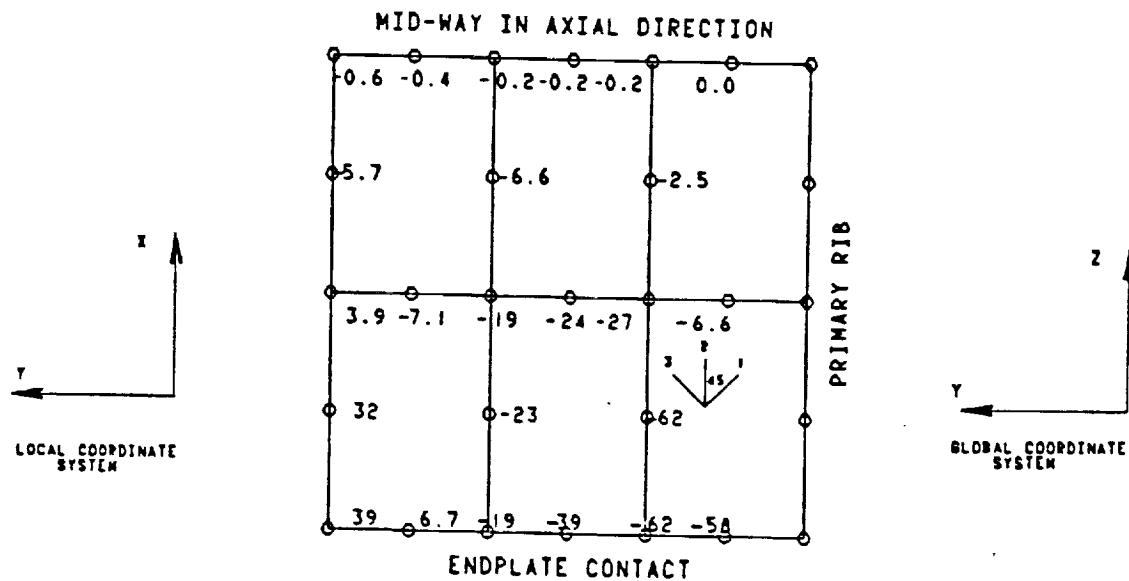
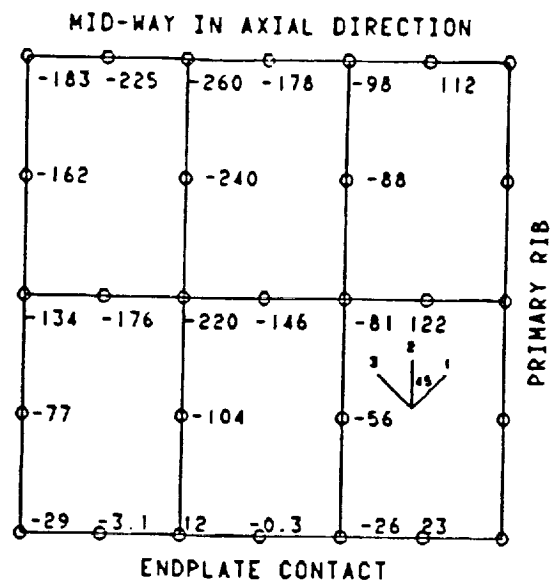
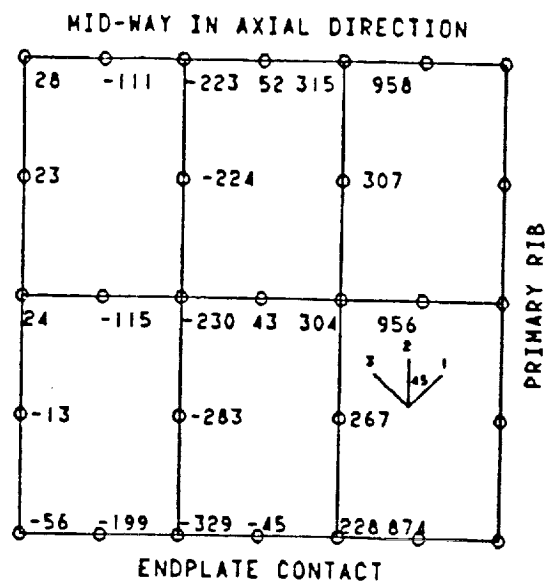
Testing of the parabolic thick shell model was completed in much the same manner as the previous models. The longitudinal tensile test was applied to the model and support case 2 again supported the model. A change in support case 2, however was required. The need for this change was described in chapter 2. Basically, in order to support axial rotation of the model, the translations of two rows of nodes were supported.

As predicted, the results of the parabolic thick shell model agreed with the previous thin shell models'. Displayed in figures 4.26-4.28, the stresses were again located in high stress gradient regions. The approximated stresses at each rosette location are listed in table 4. Comparison with the experimental data was quite good. Also, as can be seen in figure 4.29, deformation of the housing confirms the loading and support of the housing.



NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 4.26: STRESSES AT LOCATION N  
PARABOLIC THICK SHELL MODEL  
RESTRAINT CASE 2



NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 4.27: STRESSES AT LOCATION H  
PARABOLIC THICK SHELL MODEL  
RESTRAINT CASE 2



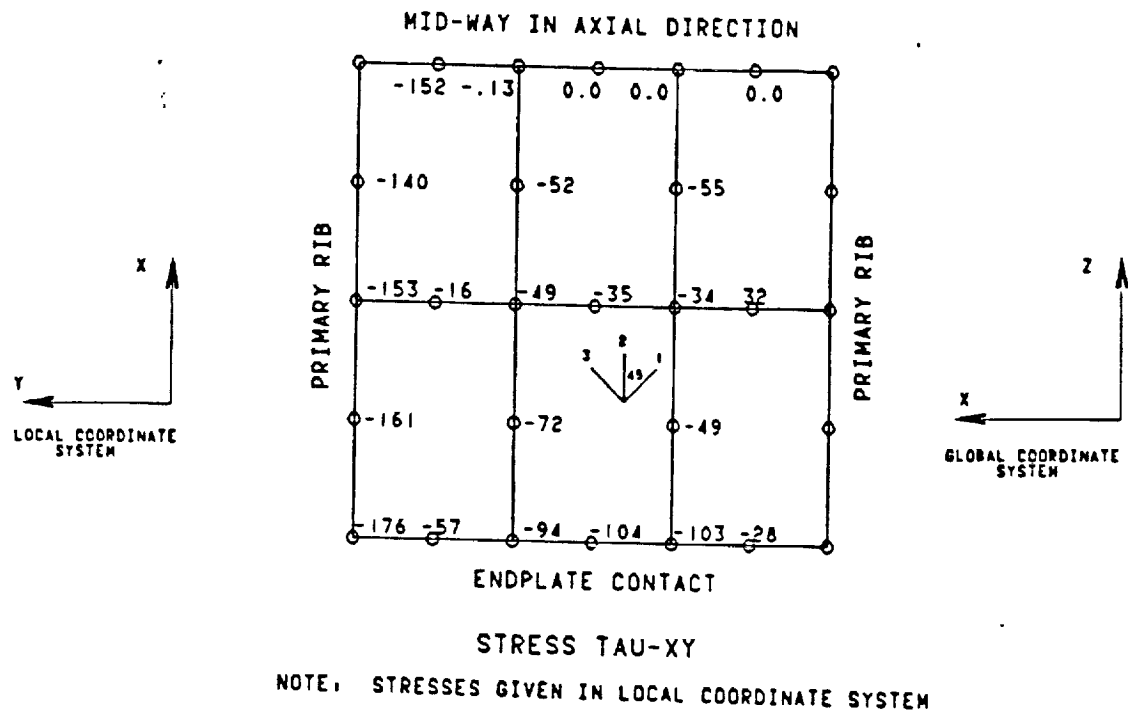
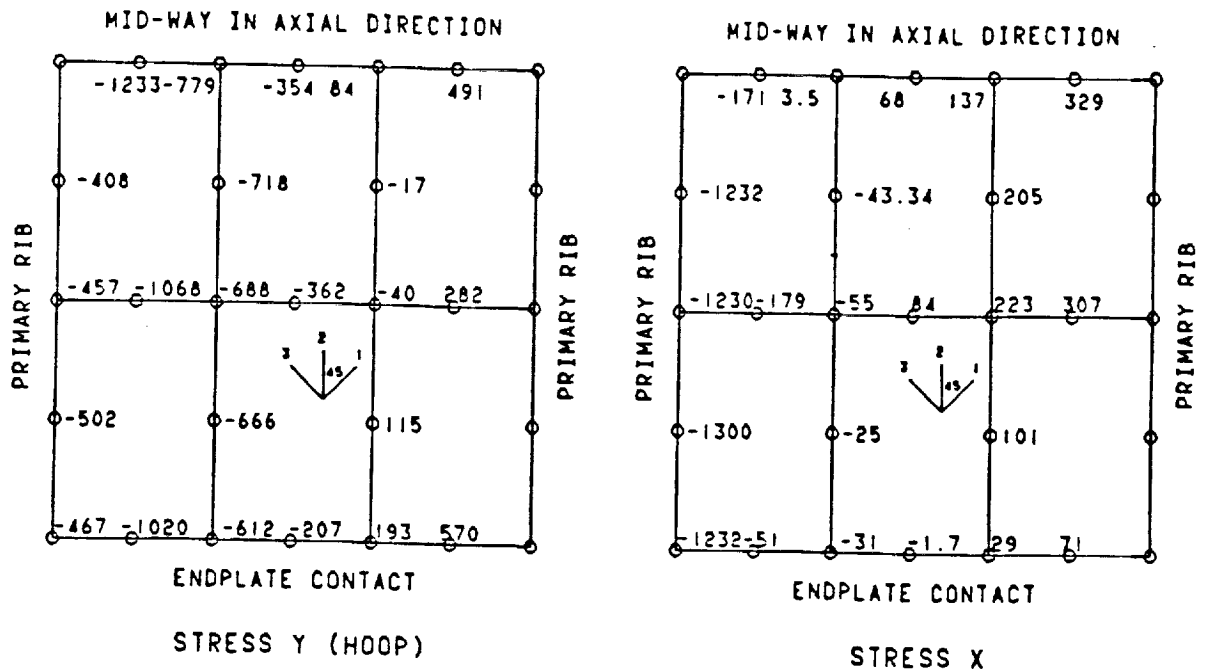


FIGURE 4.28: STRESSES AT LOCATION X  
PARABOLIC THICK SHELL MODEL  
RESTRAINT CASE 2

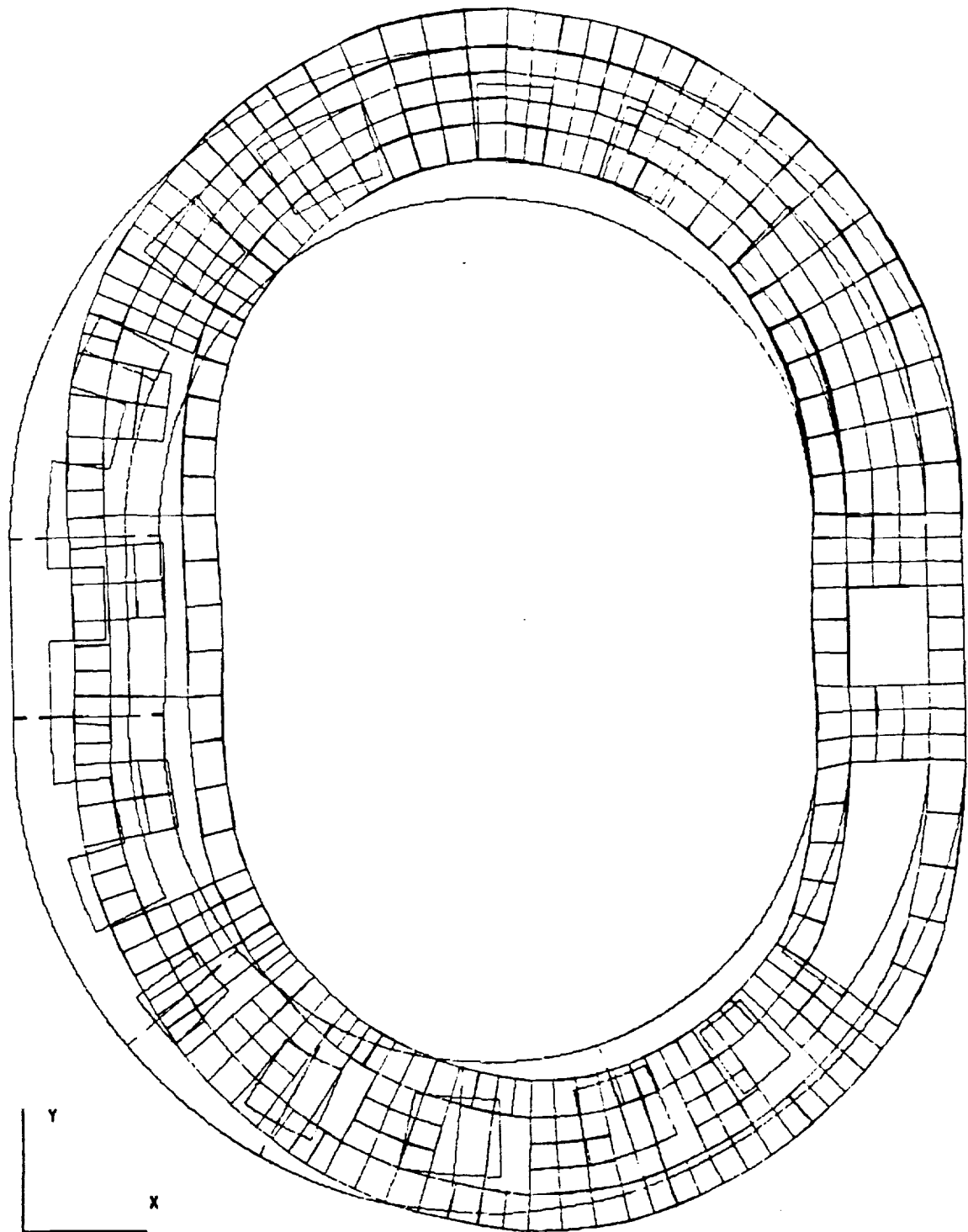


Figure 4.29: Deformation of the parabolic thick shell model subjected to a load of 1000 lbs.

LOCATION	ITEM	MODEL RESULTS	EXP DATA
N	SIGMA-X	0.0	75
	SIGMA-Y	-560	-597
	TAU-XY	-50	-85
	SIGMA-P1	4.4	86
	SIGMA-P2	-564	-608
	DIRECTION	5.1	7.1
H	SIGMA-X	50	-9.3
	SIGMA-Y	660	702
	TAU-XY	-50	36
	SIGMA-P1	664	704
	SIGMA-P2	45	-11.2
	DIRECTION	-4.7	2.9
X	SIGMA-X	90	85
	SIGMA-Y	50	100
	TAU-XY	-60	-76
	SIGMA-P1	133	170
	SIGMA-P2	6.8	15
	DIRECTION	36	48

NOTE: POSITIVE DIRECTION INDICATES  
CLOCKWISE ROTATION OF AXES.

### STRESSES GIVEN IN PSI

Table 4: Results obtained from parabolic thick shell model.

## 5.00 PRESENTATION AND REVIEW OF THE FINAL MODEL

After the analysis of the parabolic thick shell model was completed, it was clear that a structurally sound model of the housing, potentially capable of accurately completing a thermal and dynamic analysis, had been constructed. Thus, the balance of this study included adding the additional geometry previously left out: the spark plug and port regions. Furthermore, performance of this model was checked against both experimental tensile tests. Finally, since this is the foundation of a much larger study, documentation on the construction and ability of the FEM of the housing is presented.

### 5.10 CONSTRUCTION

Construction of the final FEM, shown in figure 5.1, basically added the last pieces of geometry to the parabolic thick shell model. As can be seen in figures 5.2-5.4, the placement of the ports and spark plug holes was not an easy task. Basically, three cylinders and a box were inserted radially in the model such that the inner bore and outer shell remained smooth and continuous. Placement of this geometry had to not only be consistent with the housing's geometry, but also, the meshes used had to be compatible with the existing mesh. There was no method, other than monitoring the effect on the rest of the housing, in which the construction could be verified. Thus, the introduction of the spark plugs and port regions proceeded on the basis of previous knowledge.

In order to construct the geometry of the ports and spark plug holes so that they would lie on the inner and outer surfaces, a numerical routine was again implemented (appendix 4). By utilizing this, consistent placement of the construction

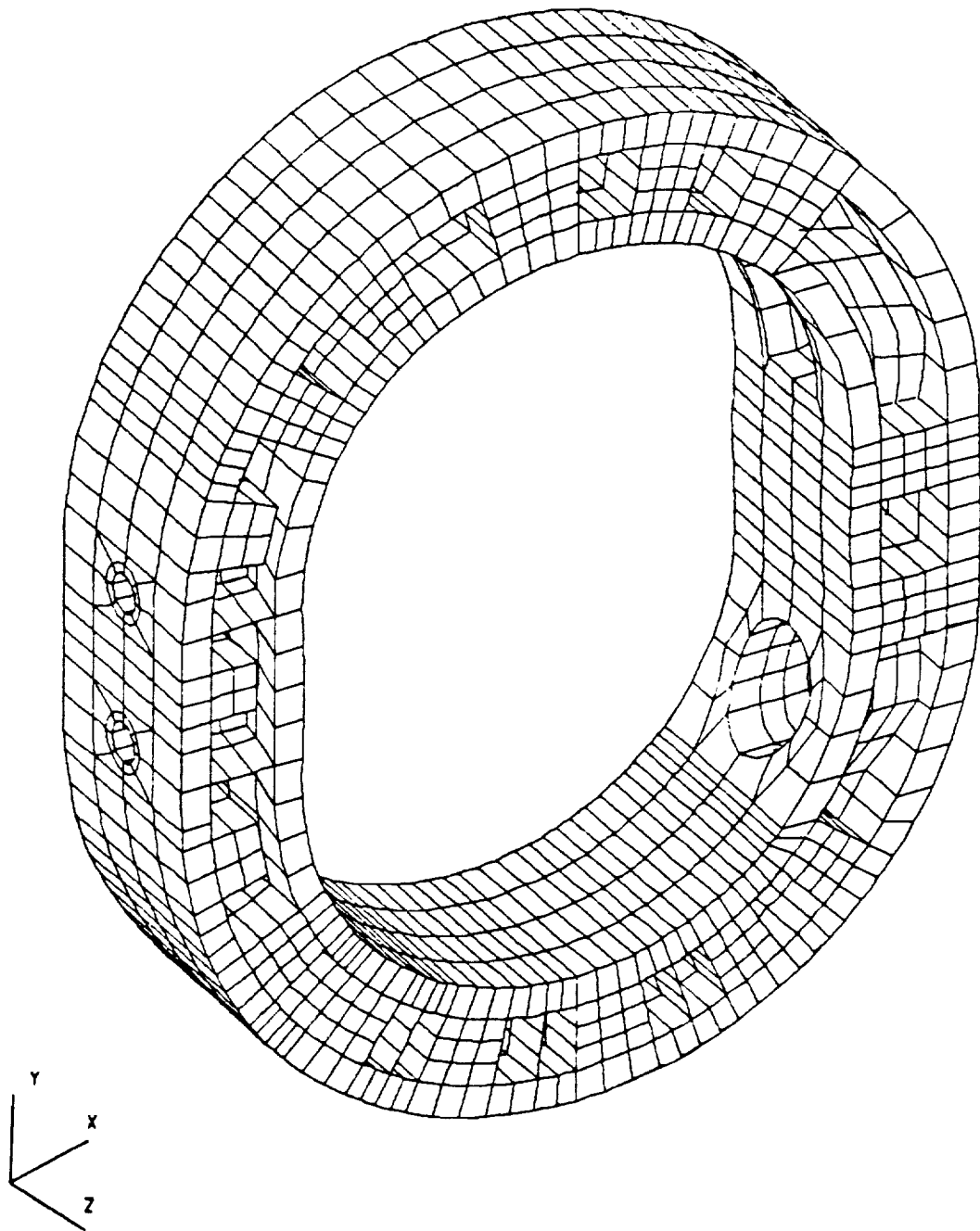


Figure 5.1: Illustration of the final FEM of the housing.

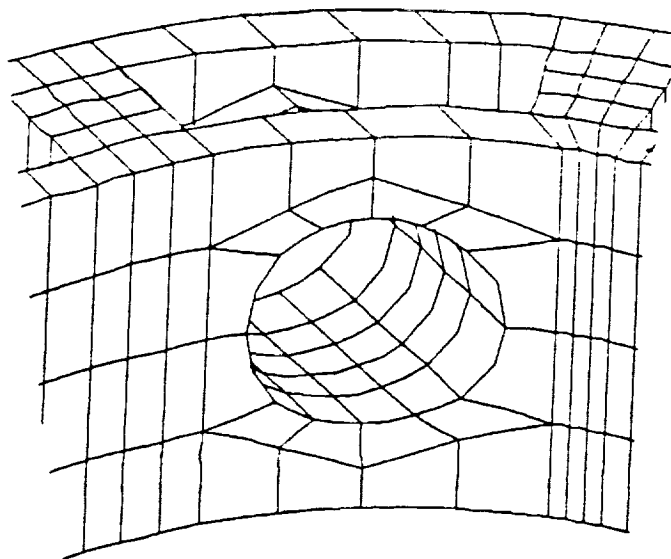


Figure 5.2: Illustration of exhaust port mesh.

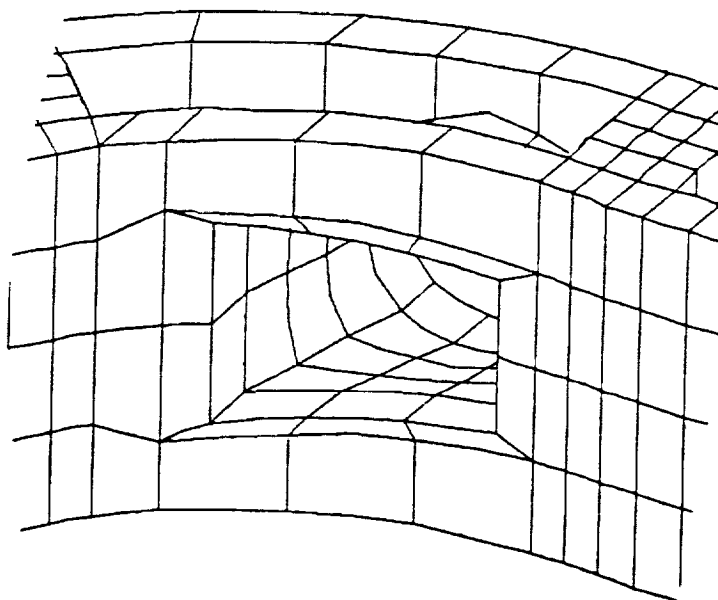
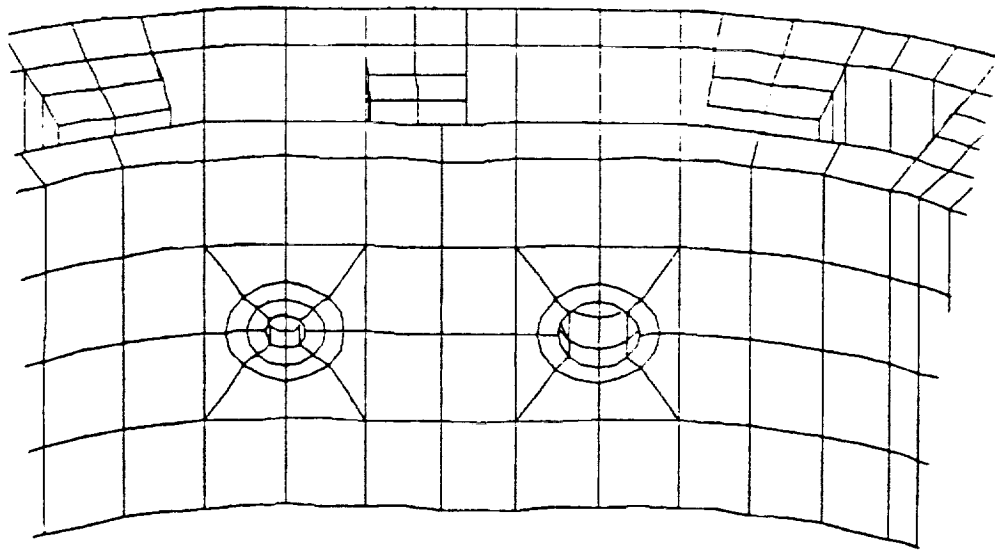
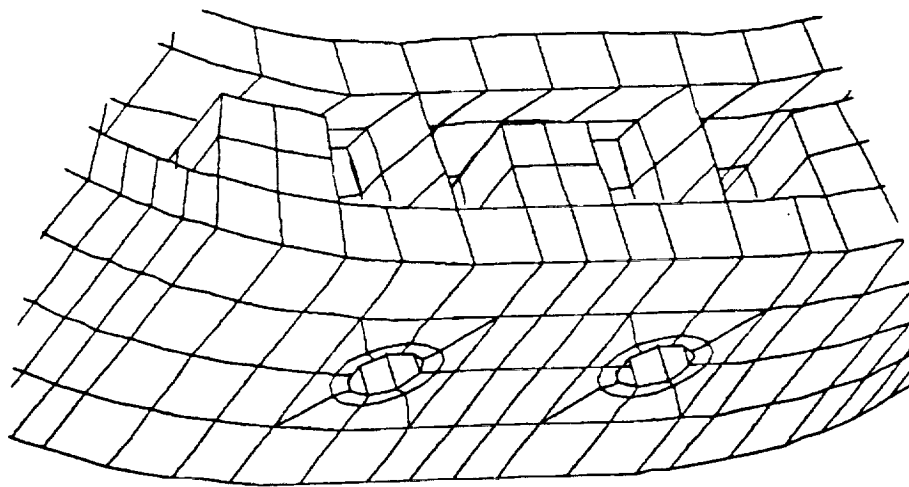


Figure 5.3: Illustration of intake port mesh.



(a) view of the inner bore.



(b) view of the outer surface.

Figure 5.4: Illustration of spark plug region.

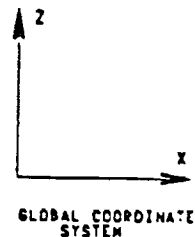
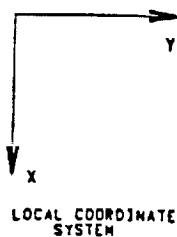
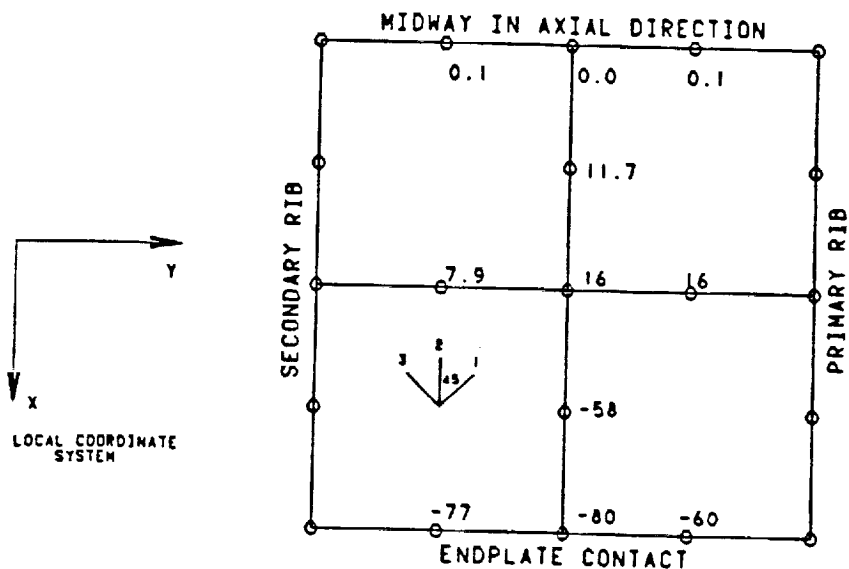
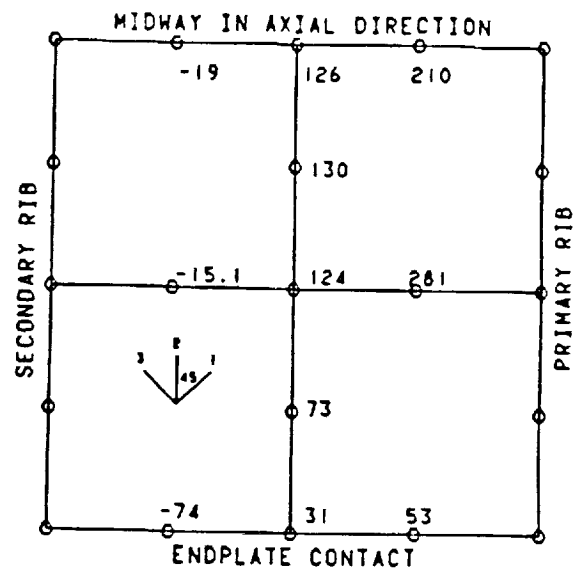
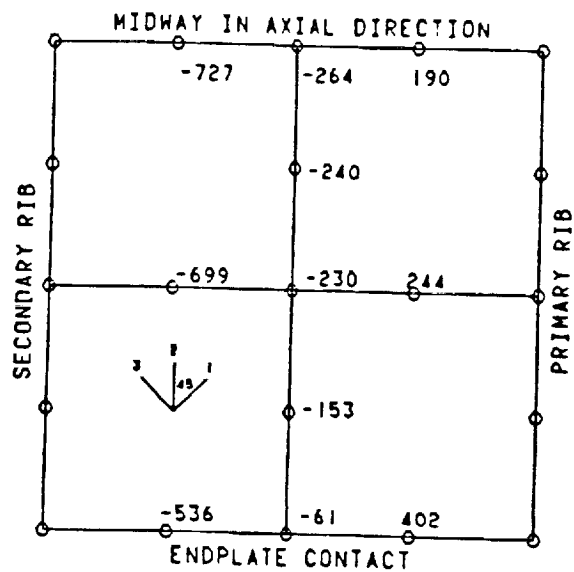
points was possible. After the wire frame geometry was completed, meshes were put in place. The resulting mesh was, again considered coarse and in actuality it consisted of a number of smaller meshes. Thick shell elements were employed for the port and spark plug walls. However, since the geometry was of odd shape, the actual connection between the ports and spark plug geometry and the housing was accomplished by using the coupling method. In these regions, though, the method did not affect the modeling capabilities at all because the elements were of the same or similar type and were therefore compatible.

In reviewing the final FEM's construction, a number of features must be noticed. First, the inner and outer surfaces were made of segments, one segment for each rib and associated cooling channel, which assured the existence of common meshes. Second, intricate geometry such as the bolt holes, rounds and fillets, and seal grooves was not included since the effect of this geometry was believed negligible. Third, there is an even number of elements in the axial direction. Fourth, elements are numbered using a radially advancing wavefront. Fifth, a coupling sequence for all coincident nodes was completed. Sixth and finally, four types of elements were used: parabolic thick shell, parabolic thin shell, parabolic solid, and linear beam.

## 5.20 PERFORMANCE AND ABILITY

Performance of the final model was, again, matched against the experimental results obtained from the longitudinal tensile test. Restraint case 2 was used to support the model. As expected, the performance of the final FEM of the housing was excellent. From the results displayed in figures 5.5-5.7 and in table 5, it can be seen that the FEM not only matched the experimental data, but also concurred with the gradients determined in the previous models. It can also be seen, in figure 5.8, that the additional geometry's affect on the housing's deformation was minor. However, since a verification





NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 5.5: STRESSES AT LOCATION N  
FINAL FEM OF THE HOUSING  
RESTRAINT CASE 2

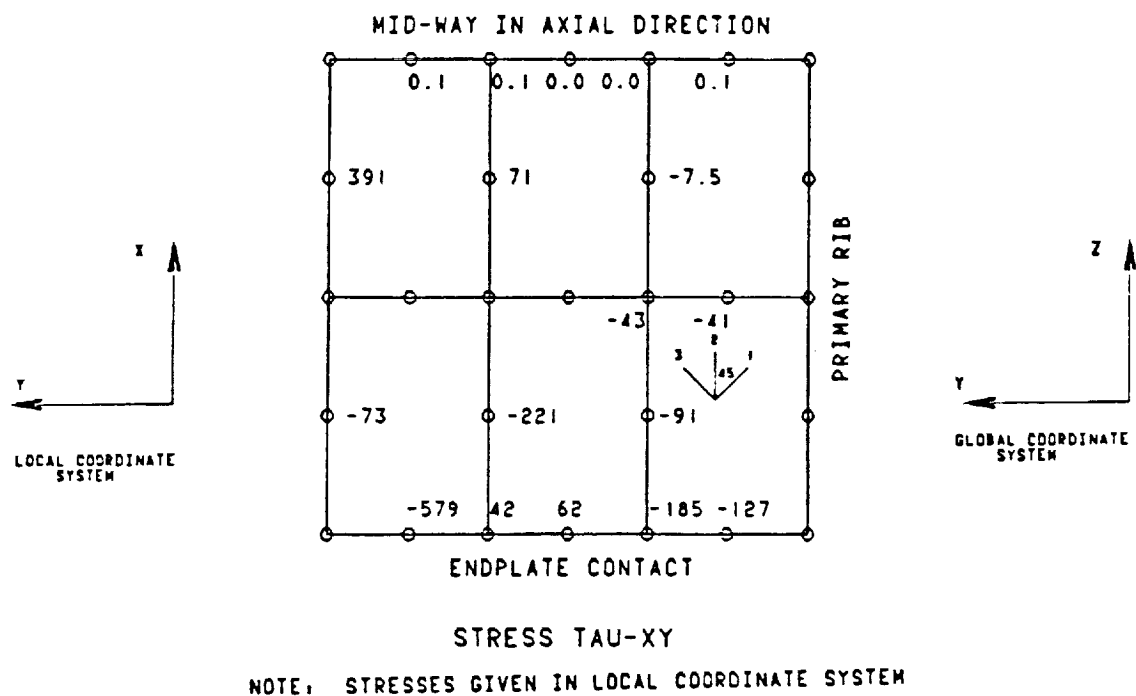
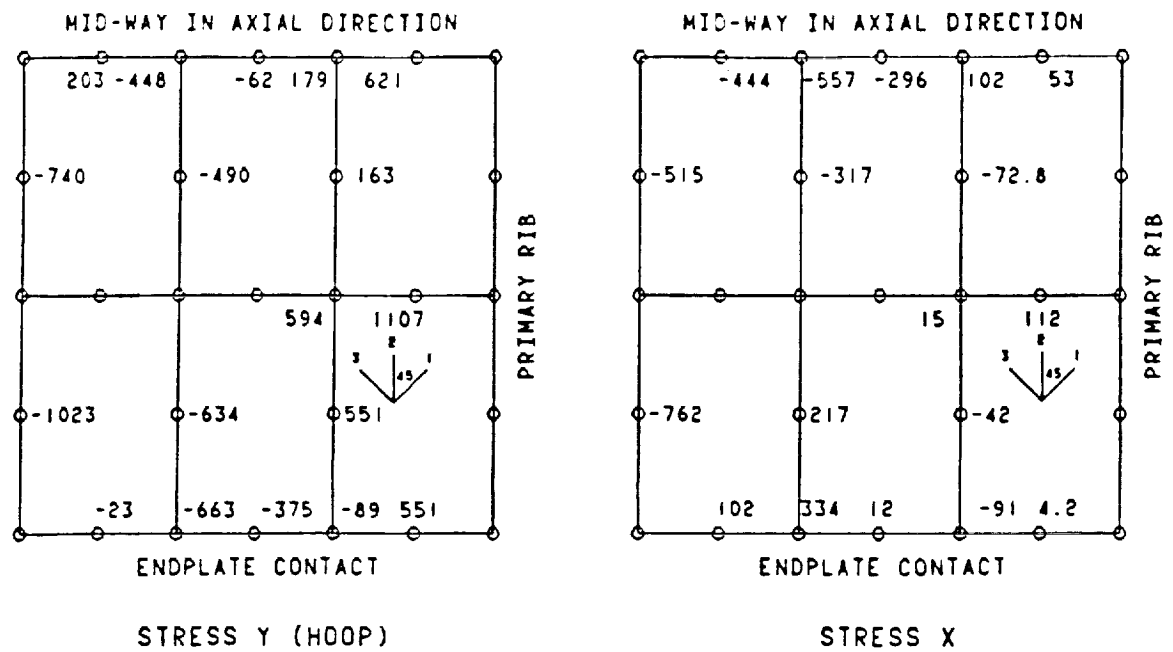
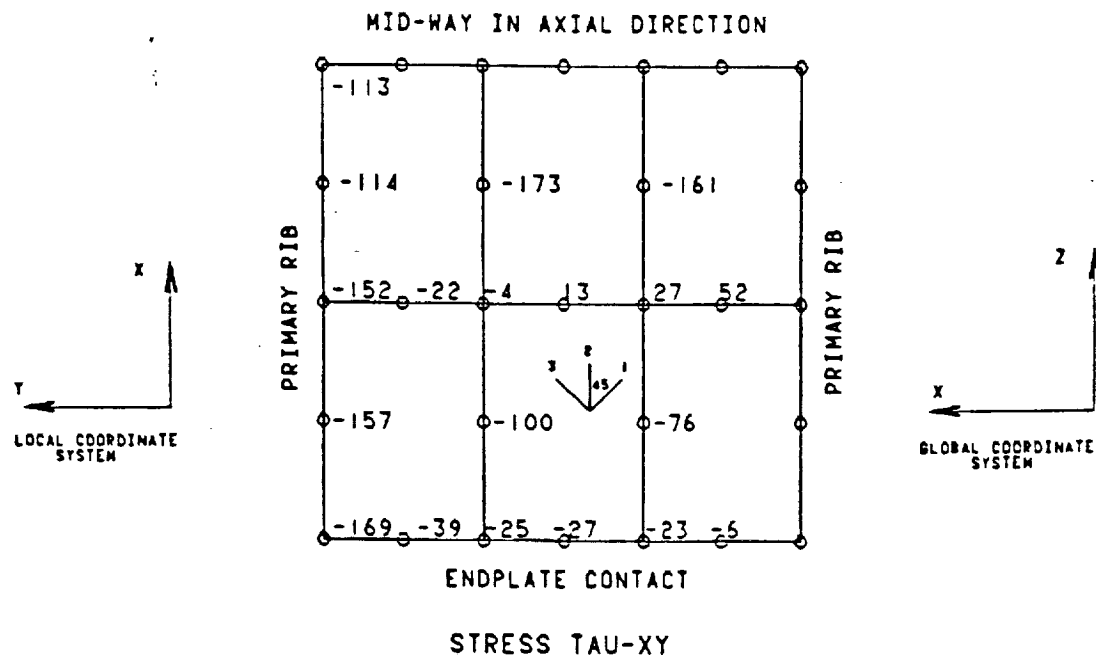
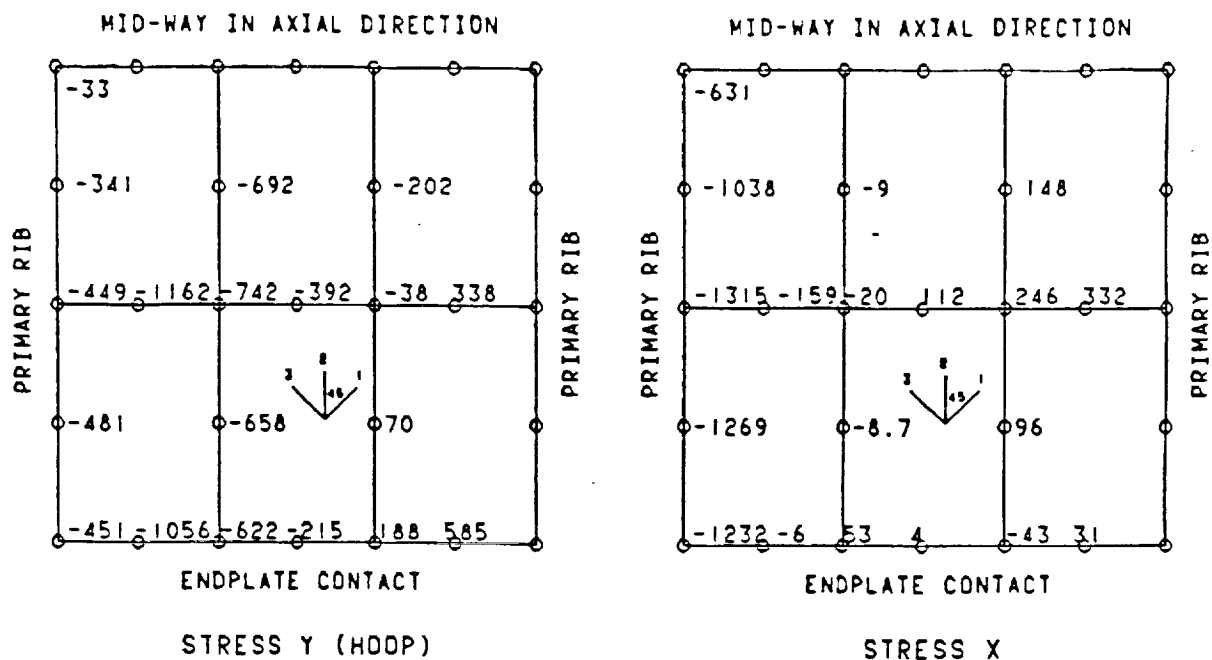


FIGURE 5.6: STRESSES AT LOCATION H  
THE FINAL FEM OF THE HOUSING  
RESTRAINT CASE 2



•NOTE: STRESSES GIVEN IN LOCAL COORDINATE SYSTEM

FIGURE 5.7: STRESSES AT LOCATION X  
FINAL FEM OF HOUSING  
RESTRAINT CASE 2

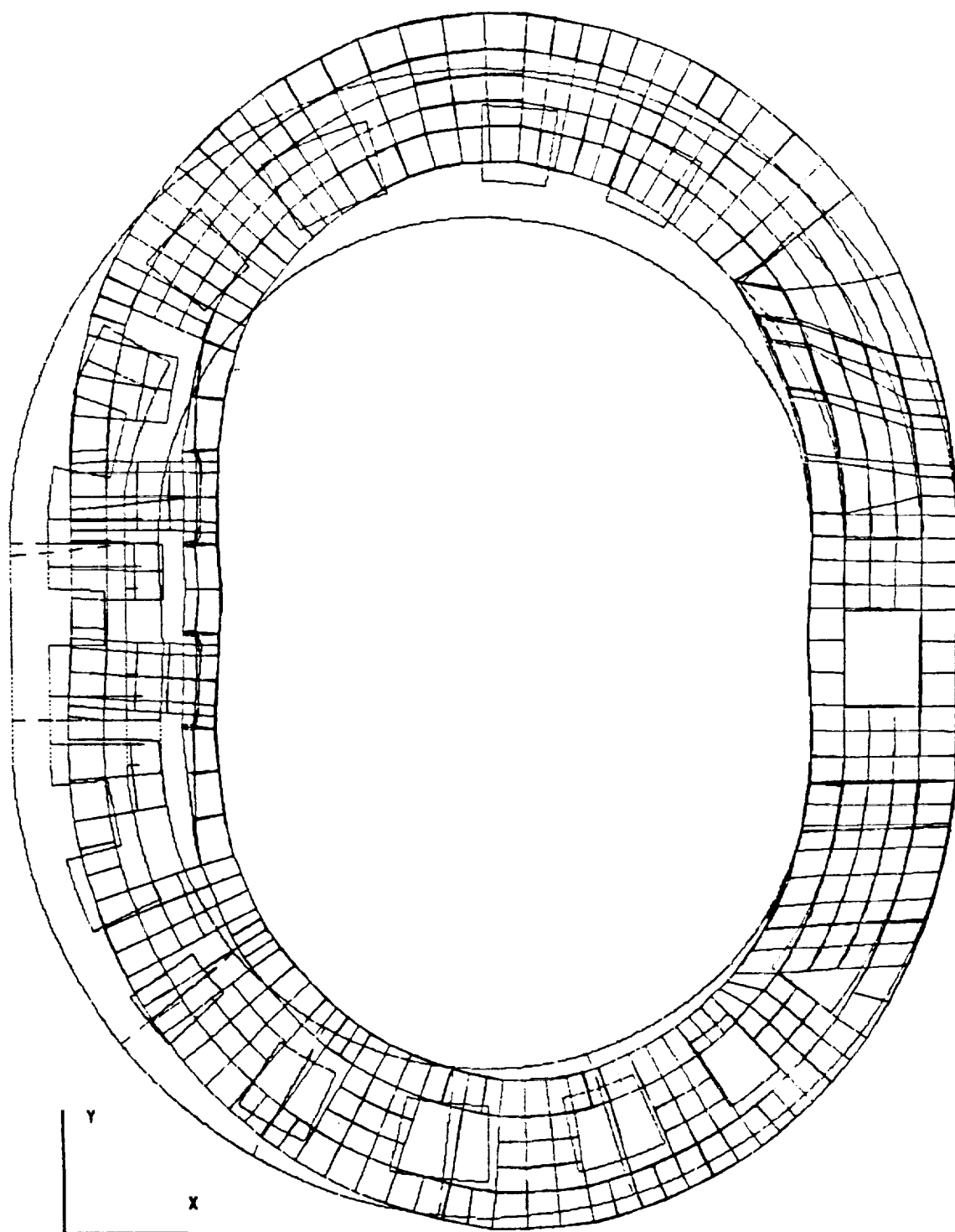


Figure 5.8: Deformation of final FEM subjected to a 1000 lb. tensile load.

LOCATION	ITEM	MODEL RESULTS	EXP. DATA
N	SIGMA-X	10	75
	SIGMA-Y	-600	-597
	TAU-XY	-45	-85
	SIGMA-P1	13.3	86
	SIGMA-P2	-603	-608
	DIRECTION	4.2	7.1
H	SIGMA-X	30	-9.3
	SIGMA-Y	600	702
	TAU-XY	-80	36
	SIGMA-P1	611	704
	SIGMA-P2	19	-11.2
	DIRECTION	-7.8	2.9
X	SIGMA-X	100	85
	SIGMA-Y	30	100
	TAU-XY	-60	-76
	SIGMA-P1	135	170
	SIGMA-P2	-4.5	15.6
	DIRECTION	30	48

NOTE: POSITIVE DIRECTION INDICATES  
CLOCKWISE ROTATION OF AXES.

STRESSES GIVEN IN PSI

Table 5: Results obtained from the final FEM.

method was not available, the deformations and stresses in those regions must be further investigated.

With these results, it was felt that the purpose of this study had been attained. A structurally sound FEM of a Wankel engine center housing had been built and verified. Although it is considered to have a coarse mesh, it by no means a small FEM. It contains over 29,000 degrees of freedom and has a maximum wavefront of 588. It consists of 516 solid, 968 thick shell, 130 thin shell, and 167 beam elements. Along with the ability to perform a static analysis, the Wankel FEM is also capable of performing a steady-state thermal and/or dynamic analysis of the housing alone. In addition, since the FEM is a SUPERTAB base file, the model can be adjusted into NASTRAN, ANSYS, or SUPERB format. With this capability, the limitations to what can be performed with this as a base model are few.

## 6.00 CONCLUSIONS AND RECOMMENDATIONS

The basic result of this study was the presentation of a finite element model of a Wankel engine center housing that is valid for static loading conditions existing on the housing alone. The model performed well and is perceived to be an excellent foundation for further study. With the graphical preprocessor SUPERTAB as a base, changes in geometry and material can now be analyzed very quickly. Further analysis must be completed to verify the thermal and dynamic capabilities of the FEM. When completing the dynamic analysis, use of the previous thin shell models might be necessary. It should be realized, though, that a dynamic analysis of the housing alone should be completed first. In addition, the construction of the exhaust port and spark plug regions should be investigated before a thermal analysis is completed. As for completing an analysis on the interaction between the endplates and the housing, the actual endplates should be modeled. The deformation of the plates as well as the clearance in the bolt holes must be considered. The internal pressure test presented in this study would provide an excellent experimental baseline for such a study. Modeling of the preload and pressurization should obtain sufficient knowledge to enable an accurate prediction of the interaction. In conclusion, it is believed that all of the studies mentioned above can be performed on the FEM presented in this thesis with very few modifications required.

## APPENDIX 1

### SUPERB FINITE ELEMENT LIBRARY

#### A1.00 STRESS OUTPUT (COORDINATE SYSTEM ORIENTATION)

From the nodal displacements computed, SUPERB determines stresses and strains at guass integration points within each element, the actual method will be discussed for each element type later. These guass point stresses are then extrapolated to the elements nodes. Thus, the nodal stress at a specific node is an average of the guass point stresses in the elements that are connected to the node. Nodal stresses are reported at each node except when a geometric discontinuity is encountered. When this occurs, guass point stresses are reported in the elements near the discontinuity. Furthermore, for thin shell elements, nodal stresses are reported at the top, middle, and bottom surfaces at each node. Typical stress outputs are shown in figures A1.1, A1.2, and A1.3.

Stresses can be reported in at least three different coordinate systems: global, local, or elemental. The global system refers to the coordinate system in which the geometry and nodal coordinates are defined. The local and elemental coordinate systems are defined by each element. Nodal stresses are reported in the global (G-STs), or local (L-STs), coordinate systems while the guass point stresses are reported in the elemental coordinate system. Exceptions to this are found when two different element types are connected (9).

VERSION 6.31 (RELEASE 01FEB84) IBM 02/ 8/85 12:25:52 C A E D S - S U P E R B FINITE ELEMENT ANALYSIS												STATIC ANALYSIS	
LOADING CASE 2 (SEQUENCE 1) MESH THICK SHELL MODEL WITH ADDED GAO TRANS. LOAD													
NODE #STS	STRESS-X	STRESS-Y	STRESS-Z	STRESS-XZ	STRESS-YZ	STRESS-ZZ	*****PRINCIPAL STRESSES*****					VON MISES	
9711 L-SYS	-23.93	1.794	-80.41	.0	.0	.0	-23.77	-80.47	.0	.0	.0	71.60	
---	42.97	-2.258	34.10	.0001	50.23	.00	75.04	-41.54	.0	.0	.0	102.1	
---	42.97	-1.293	149.0	.0	.0	.0	144.0	42.95	.0	.0	.0	132.9	
9713 L-SYS	15.13	2.733	144.8	.0	.0	.0	144.7	15.08	.0	.0	.0	157.7	
---	-22.87	1.501	34.09	-7407	50.20	.00	74.19	-41.51	-22.85	.0	.0	102.1	
---	-61.07	-2.713	-98.53	.0	.0	.0	-61.07	-98.54	.0	.0	.0	84.50	
9715 L-SYS	-53.96	-5.963D-02	-20.4	.0	.0	.0	-53.96	-20.40	.0	.0	.0	183.1	
---	20.32	-1.253	38.42	.0372	50.75	.00	78.19	-40.50	.0	.0	.0	104.4	
---	100.6	-3.266	281.3	.0	.0	.0	281.9	100.5	.0	.0	.0	246.6	
9716 L-SYS	-27.91	.8492	-74.41	.0	.0	.0	-27.90	-74.41	.0	.0	.0	88.84	
---	11.50	-1.951	37.83	.0	.0	.0	37.83	11.50	.0	.0	.0	49.74	
---	50.91	-4.450	154.2	.0	.0	.0	154.4	50.71	.0	.0	.0	136.3	
9717 L-SYS	-2.540	.6914	61.94	.0	.0	.0	61.95	-2.540	.0	.0	.0	63.27	
---	-6.975	-1.174	33.74	.0124D-02	50.01	.00	73.40	-41.19	.0	.0	.0	103.4	
---	-6.810	-2.879	25.82	.0	.0	.0	25.87	-7.064	-6.671	.0	.0	30.33	
9718 L-SYS	22.48	3.051	162.3	.0	.0	.0	162.1	22.48	.0	.0	.0	182.0	
---	-22.48	-1.053	20.42	.0933	51.43	.00	70.44	-41.19	.0	.0	.0	103.4	
---	-64.49	-1.9517	-103.0	.0	.0	.0	-64.44	-103.1	-20.78	.0	.0	103.4	
9719 L-SYS	45.78	4.657	288.4	.0	.0	.0	288.0	45.67	.0	.0	.0	288.1	
---	-35.00	3.707	24.52	.0975	50.00	.00	73.70	-41.19	.0	.0	.0	103.4	
---	-123.8	2.756	-229.7	.0	.0	.0	-123.7	-229.8	-30.57	.0	.0	149.2	
9721 L-SYS	-19.26	5.248	-82.45	.0	.0	.0	-18.83	-83.38	.0	.0	.0	75.74	
---	34.52	-6.452	34.38	.0274	50.25	.00	82.38	-43.31	.0	.0	.0	109.8	
---	34.52	-18.55	161.1	.0	.0	.0	163.5	24.58	.0	.0	.0	152.7	
9723 L-SYS	18.16	6.009	175.7	.0	.0	.0	178.9	17.94	.0	.0	.0	170.4	
---	-35.10	-1.259	-40.32	.0218	50.27	.00	73.40	-43.18	.0	.0	.0	103.4	
---	-35.10	-1.259	-40.32	.0	.0	.0	-35.10	-40.32	-6.826	.0	.0	42.44	
9725 L-SYS	-9.131	6.352	-224.9	.0	.0	.0	-8.924	-225.1	.0	.0	.0	220.8	
---	25.181	-1.109	245.1	.0449	50.23	.00	83.50	-43.24	.0	.0	.0	119.4	
---	25.181	-4.314	245.1	.0	.0	.0	245.2	25.35	.0	.0	.0	243.1	
9726 L-SYS	-7.428	7.476	-55.44	.0	.0	.0	-6.660	-55.21	.0	.0	.0	80.07	
---	-2.902	-7.476	16.13	.0756	50.54	.00	71.46	-43.74	.0	.0	.0	117.4	
---	-2.902	-7.476	16.13	.0	.0	.0	-2.902	-16.13	-4.686	.0	.0	171.1	
9727 L-SYS	.5031	12.43	5.71	.0	.0	.0	.57	12.138	.0	.0	.0	58.1	
---	.5031	12.43	5.71	.0065	50.54	.00	57.01	-42.60	.0	.0	.0	42.71	
---	.5031	12.43	5.71	.0	.0	.0	.5031	12.43	-12.65	.0	.0	42.71	

VERSION 6.31 (RELEASE 01FEB84) IBM 02/ 8/85 12:25:52 C A E D S - S U P E R B FINITE ELEMENT ANALYSIS												STATIC ANALYSIS	
												PAGE 27	

Figure A1.1: Typical thin shell element stress output.

VERSION 6.31 (RELEASE 01FEB84) IBM 02/ 8/85 12:25:52										C A E D S - S U P E R B FINITE ELEMENT ANALYSIS										STATIC ANALYSIS									
LOADING CASE 2 (SEQUENCE 1)										MESH THICK SHELL MODEL WITH ADDED GAO TRANS. LOAD																			
NODE #	STS	STRESS-X	STRESS-Y	STRESS-Z	STRESS-XZ	STRESS-YZ	STRESS-ZZ	*****PRINCIPAL STRESSES*****					VON MISES																
9711	L-SYS	-23.93	1.794	-80.41	.0	.0	.0	-23.77	-80.47	.0	.0	.0	71.60																
---	---	42.97	-2.258	34.10	.0001	50.23	.00	75.04	-41.54	.0	.0	.0	102.1																
---	---	42.97	-1.293	149.0	.0	.0	.0	144.0	42.95	.0	.0	.0	132.9																
9713	L-SYS	15.13	2.733	144.8	.0	.0	.0	144.7	15.08	.0	.0	.0	157.7																
---	---	-22.87	1.501	34.09	-7407	50.20	.00	74.19	-41.51	-22.85	.0	.0	102.1																
---	---	-61.07	-2.713	-98.53	.0	.0	.0	-61.07	-98.54	.0	.0	.0	84.50																
9715	L-SYS	-53.96	-5.963D-02	-20.4	.0	.0	.0	-53.96	-20.40	.0	.0	.0	183.1																
---	---	20.32	-1.253	38.42	.0372	50.75	.00	78.19	-40.50	.0	.0	.0	104.4																
---	---	100.6	-3.266	281.3	.0	.0	.0	281.9	100.5	.0	.0	.0	246.6																
9716	L-SYS	-27.91	.8492	-74.41	.0	.0	.0	-27.90	-74.41	.0	.0	.0	88.84																
---	---	11.50	-1.951	37.83	.0	.0	.0	37.83	11.50	.0	.0	.0	49.74																
---	---	50.91	-4.450	154.2	.0	.0	.0	154.4	50.71	.0	.0	.0	136.3																
9717	L-SYS	-2.540	.6914	61.94	.0	.0	.0	61.95	-2.540	.0	.0	.0	63.27																
---	---	-6.975	-1.174	33.74	.0124D-02	50.01	.00	73.40	-41.19	.0	.0	.0	103.4																
---	---	-6.810	-2.879	25.82	.0	.0	.0	25.87	-7.064	.0	.0	.0	30.33																
9718	L-SYS	22.48	3.051	162.3	.0	.0	.0	162.1	22.48	.0	.0	.0	182.0																
---	---	-22.48	-1.053	20.42	.0933	51.43	.00	70.44	-41.19	.0	.0	.0	103.4																
---	---	-64.49	-1.9517	-103.0	.0	.0	.0	-64.44	-103.1	.0	.0	.0	103.4																
9719	L-SYS	45.78	4.657	288.4	.0	.0	.0	288.0	45.67	.0	.0	.0	288.1																
---	---	-35.00	3.707	24.52	.0975	50.00	.00	73.70	-41.19	.0	.0	.0	103.4																
---	---	-123.8	2.756	-229.7	.0	.0	.0	-123.7	-229.8	.0	.0	.0	149.2																
9721	L-SYS	-19.26	5.248	-82.45	.0	.0	.0	-18.83	-83.38	.0	.0	.0	75.74																
---	---	34.52	-6.452	34.38	.0274	50.25	.00	82.38	-43.31	.0	.0	.0	109.8																
---	---	34.52	-18.55	161.1	.0	.0	.0	163.5	24.58	.0	.0	.0	152.7																
9723	L-SYS	18.16	6.009	175.7	.0	.0	.0	178.9	17.94	.0	.0	.0	170.4																
---	---	-35.10	-1.259	-40.32	.0218	50.27	.00	73.40	-43.18	.0	.0	.0	103.4																
---	---	-35.10	-1.259	-40.32	.0	.0	.0	-35.10	-40.32	.0	.0	.0	42.44																
9725	L-SYS	-9.131	6.352	-224.9	.0	.0	.0	-8.924	-225.1	.0	.0	.0	220.8																
---	---	25.181	-1.109	245.1	.0449	50.23	.00	83.50	-43.24	.0	.0	.0	119.4																
---	---	25.181	-4.314	245.1	.0	.0	.0	245.2	25.35	.0	.0	.0	243.1																
9726	L-SYS	-7.428	7.476	-55.44	.0	.0	.0	-6.660	-55.21	.0	.0	.0	80.07																
---	---	-2.902	-7.476	16.13	.0756	50.54	.00	71.46	-43.74	.0	.0	.0	117.4																
---	---	-2.902	-7.476	16.13	.0	.0	.0	-2.902	-16.13	.0	.0	.0	171.1																
9727	L-SYS	.5031	12.43	5.71	.0	.0	.0	.57	12.138	.0	.0	.0	58.1																
---	---	.5031	12.43	5.71	.0065	50.54	.00	57.01	-42.60	.0	.0	.0	42.71																
---	---	.5031	12.43	5.71	.0	.0	.0	.5031	12.43	.0	.0	.0	42.71																

VALUES ARE IN INCH D. VALS IN GAUSS. PRINTS ARE ROUNDED.

CAE D S - S U P E R B  
FINITE ELEMENT ANALYSIS

PAGE 000001  
STATIC ANALYSIS

Figure A1.2: Typical thick shell element stress output.



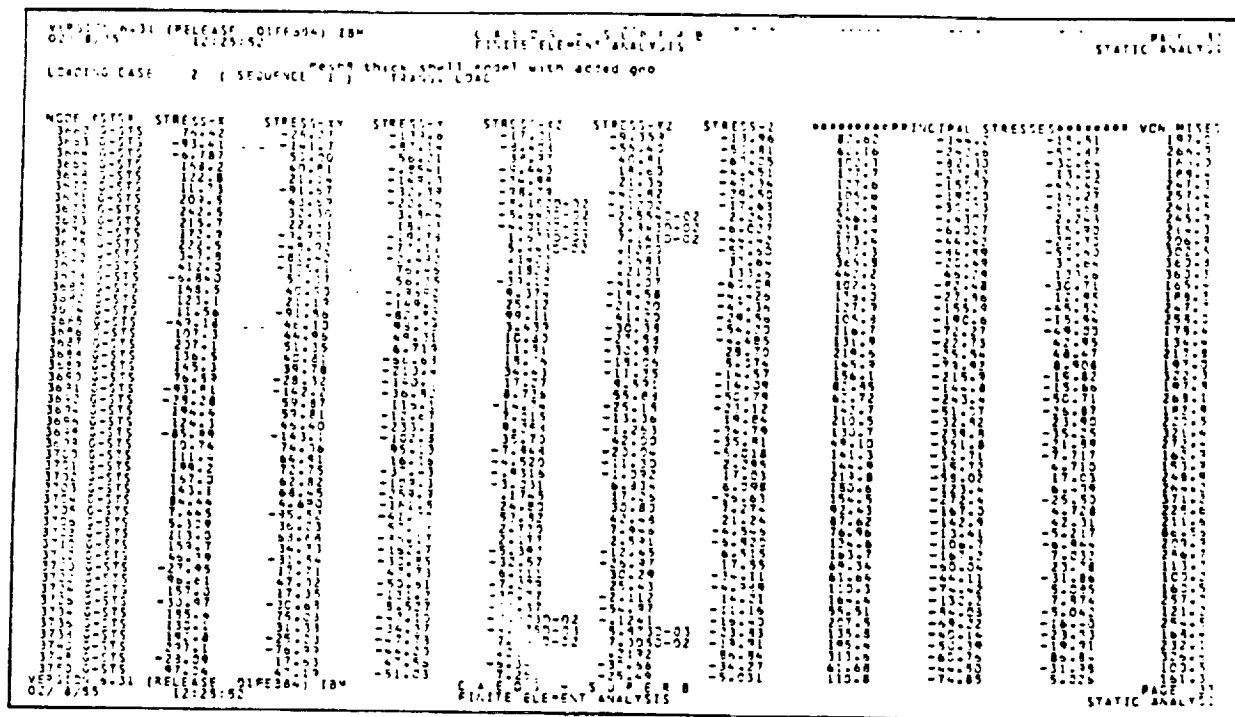


Figure A1.3: Typical solid element stress output.

#### A1.10 THIN SHELL ELEMENT

As seen in figure A1.4, three basic nodal configurations are available with the thin shell: linear (4-node), parabolic (8-node), and cubic (12-node). Various transition elements are also available to aid in modeling. Each node of a thin shell element is assigned six degrees of freedom (3 translations and 3 rotations) and is used to define the element's middle surface. Thus, the element is capable of modeling bending and membrane actions. Also, the parabolic and cubic elements are capable of modeling deformations due to shearing forces, whereas the linear elements are not. In addition to this, each element can be assigned isotropic or orthotropic material properties. In computations the stress normal to the element's middle surface ( $\sigma_z$ ) is assumed zero. Nodal stresses are reported at the top, middle, and bottom surfaces at each node. The top and bottom surfaces of a thin shell element are defined by the element connectivity as shown in figure A1.5.

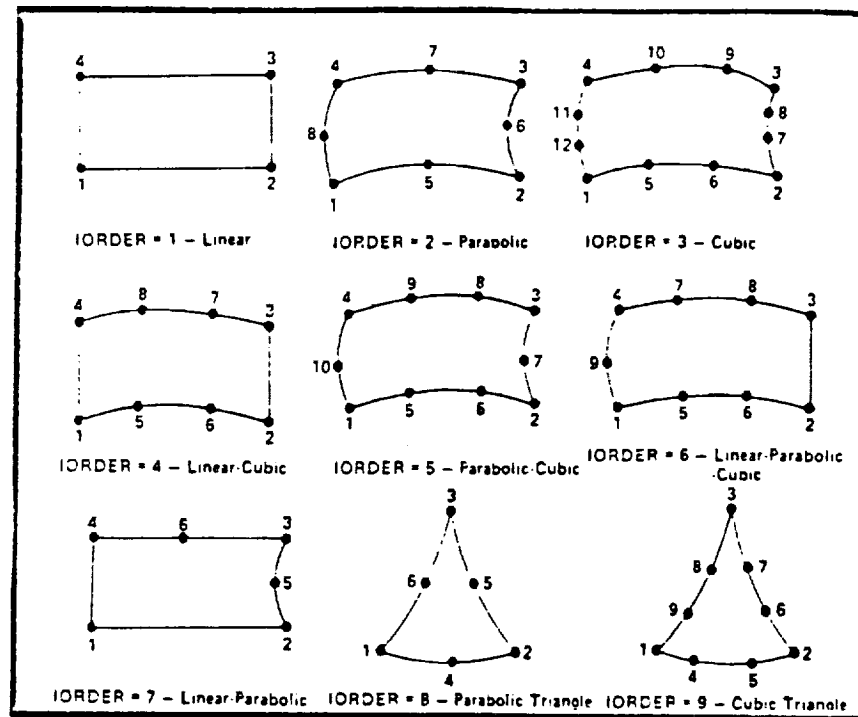


Figure A1.4: Various thin shell elements available.  
(Courtesy of SDRC)

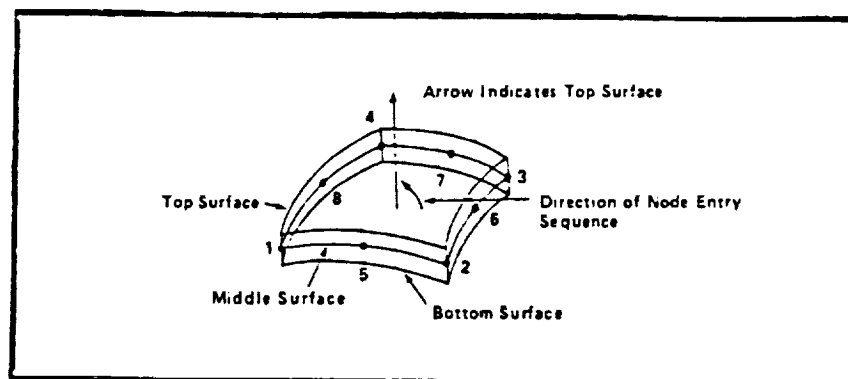


Figure A1.5: Thin shell surface definition.  
(Courtesy of SDRC)

The thermal capability of the thin shell element is limited to in-plane steady state temperature distributions. Heat conduction prepindicular to the element's middle surface is assumed zero. Although each surface can have its own convective coefficient and ambient temperature, convective heat transfer through each surface uses the same body temperature. Furthermore, heat transfer through the element's free edges is assumed zero.

Element connectivity determines the local and elemental coordinates systems. In both systems the Z-axis is normal to the shell's surface and in the direction of the top surface. The local X-axis is defined by crossing the global Y-axis and the local Z-axis. If the angle between the local Z-axis and the global Y-axis is less than 18 degrees, the global X-axis is used to determine the local X-axis. Once the local Z and X axes are determined the local Y-axis is found using the right-hand rule. The process is illustrated in figure A1.6.

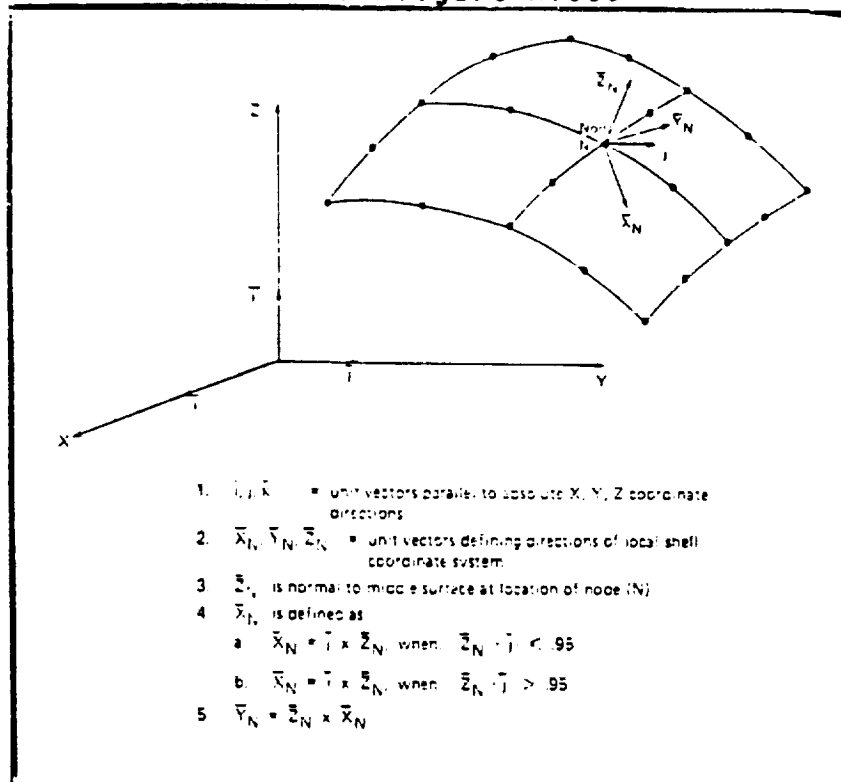


Figure A1.6: Local axis orientation.

(Courtesy of SDRC)

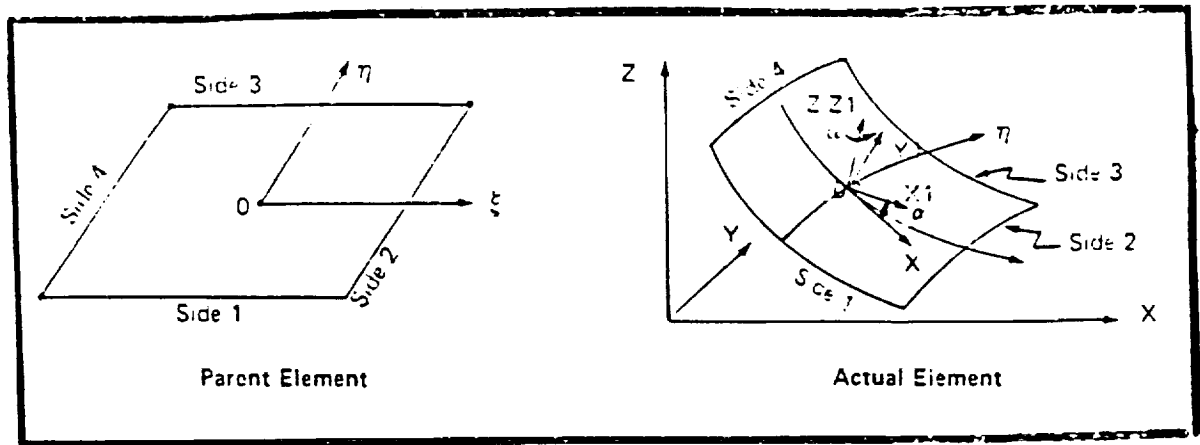


Figure A1.7: Elemental axis orientation.  
(Courtesy of SDRC)

Continuing, the elemental X-axis is tangent to the element's middle surface and perpendicular to its second side as defined by the element's connectivity. The element Y-axis is also tangent to the element's middle surface but is perpendicular to its third side. The elemental coordinate system is displayed in figure A1.7.

#### A1.20 THICK SHELL ELEMENT

As seen in figure A1.8, the thick shell element is constructed using brick type geometry. Each node of the element is assigned three degrees of freedom, all translational, and as in the thin shell element the stress normal to the element's middle surface ( $\sigma_z$ ) is assumed zero. In addition, as in the thin shell element, the thick shell elements can be assigned isotropic or orthotropic material properties. The nodal configurations which are available include the linear (8-node), parabolic (16-node), and cubic (24-node), and various transition

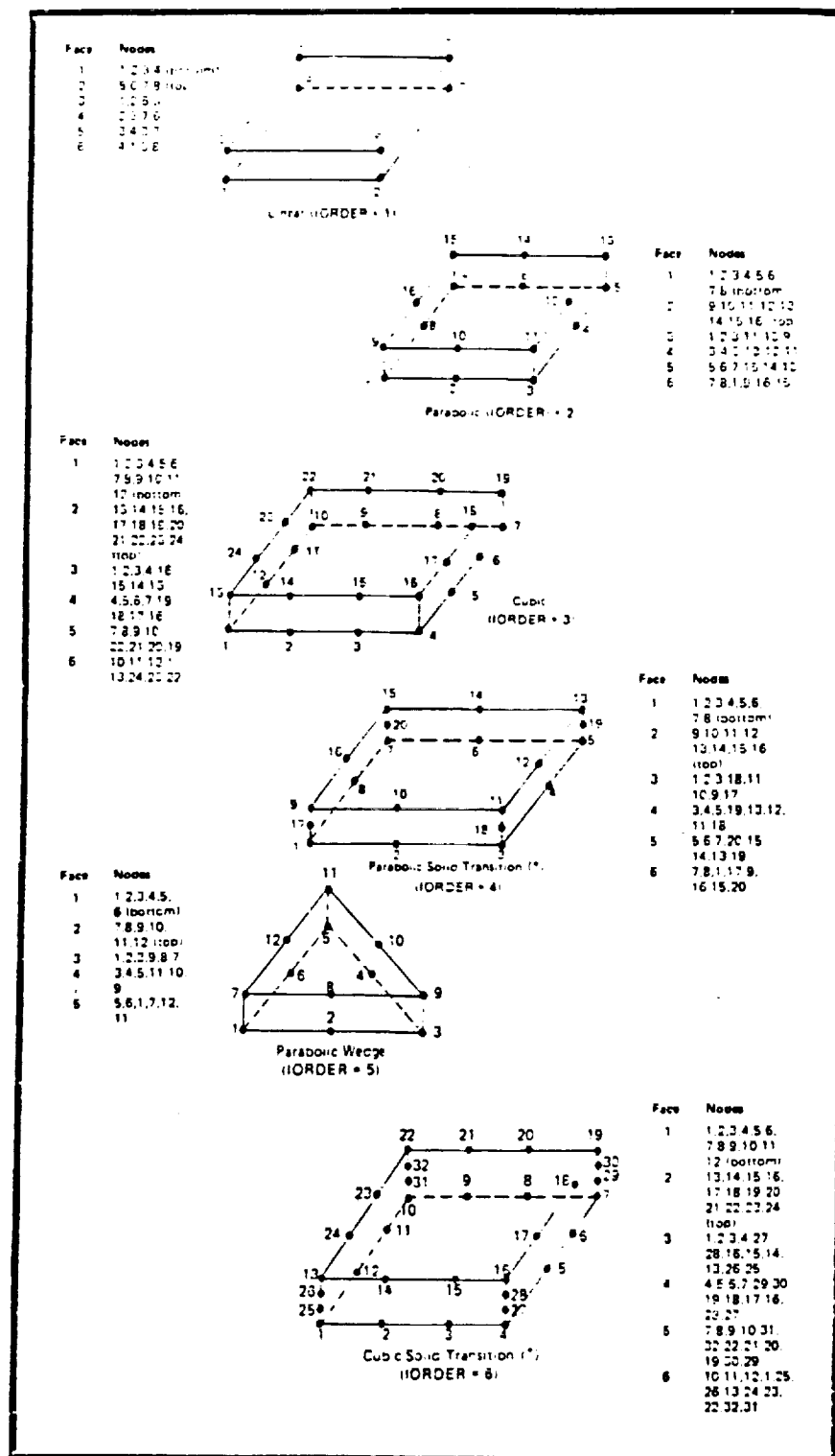


Figure A1.8: Various thick shell elements available.  
(Courtesy of SDRC)

elements. It is capable of modeling membrane, bending and shearing actions. Furthermore, heat transfer can occur through any of its six faces. Thus, the thick shell element is capable of modeling a steady-state temperature distribution through its thickness. As with the thin shell elements, the top and bottom surfaces are defined by the element connectivity as illustrated in figure A1.9. The local and elemental coordinate systems are also defined in a similar manner as the thin shell element. Nodal stresses are given in the local coordinate system (exceptions are given in reference 9).

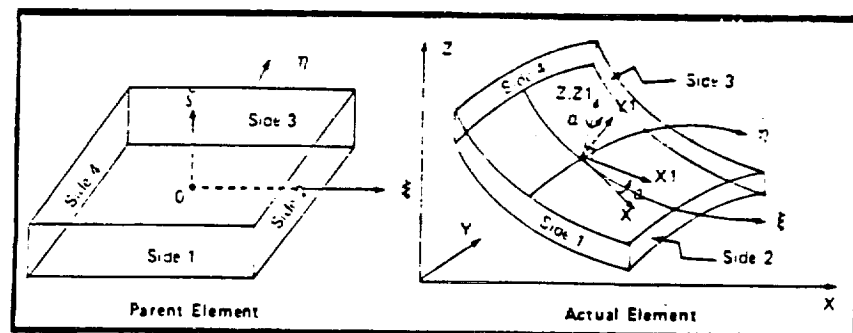


Figure A1.9: Thick shell element surface definition.  
(Courtesy of SDRC)

### A1.30 SOLID ELEMENT

As with the thick shell element, the solid element, shown in figure A1.10, uses 3-D "brick" geometry to describe itself. Each node is assigned 3 degrees of freedom, all translational, and can use isotropic or orthotropic material properties. The solid element, as determined by the benchmark study, is fully capable of modeling uniaxial loading. However, a number of elements are required for adequate modeling of bending actions. In addition,

the solid element can model thermal distributions extremely well, since heat transfer can occur through each face of the element.

#### A1.40 BEAM ELEMENT

The final element type utilized was the beam element. Although each beam element possesses the ability of being linear, tapered, or curved, only linear beam elements were used in the modeling of the housing. Each node of the beam element supports 6 degrees of freedom (3 translation and 3 rotation). Thus, it is compatible with the thin shell element. By specifying no cross sectional area and large moments of inertia, a beam, extremely rigid to bending forces but able to offer no resistance to uniaxial loading can be obtained. Thus, the beam element was used primarily in stiffening the areas around the secondary ribs and mid-plane stiffeners. In conclusion, even though thermal capabilities are available, utilization of them was not necessary.

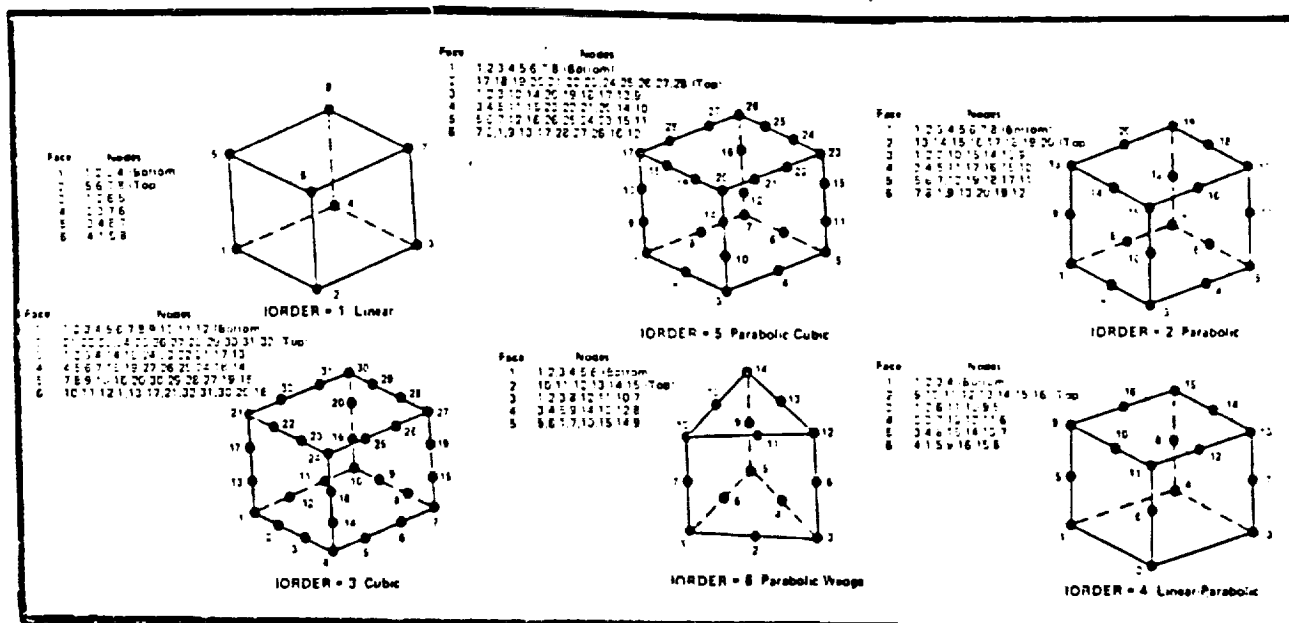


Figure A1.10: Various solid elements available.

(Courtesy of SDRC)

## APPENDIX 2

### ELEMENT BENCHMARK TESTS

In order to determine the capabilities of the element types available to model the housing, a benchmark study was completed for each type of element. Tests were devised to exercise the element's under loading conditions similar to those expected in the housing. Since the housing would be subjected to bending, membrane, and shearing actions, the shell elements were subjected to these types of loadings and their capabilities were determined. Various mesh sizes and aspect ratios were used and exact solutions were found to determine convergence and accuracy.

#### A2.00 SHELL ELEMENT TESTS

Two tests were devised to exercise the shell elements available. The flat plate was constructed as a thick, short plate to introduce a shearing action of substantial magnitude. Both tests were constrained on two opposing sides while the remaining two sides were left free.

An exact solution was found in reference (10) (see appendix 3) for a flat plate simply-supported on two opposite sides and free on the remaining two sides. Due to the way the thick shell elements are constructed, though, only the thin shell elements were tested against the exact solution. The simply-supported sides were then clamped and the thick shell elements were tested against the thin shell elements. This method insured that the same boundary conditions were applied to each element type. Furthermore, the exact solution did not account for deformation



due to shear so a third test was devised to decrease the effect of the shearing action. This was done by decreasing the thickness of the plate. The tests are illustrated in figures A2.1, A2.2 and the results are given in tables 6 and 7.

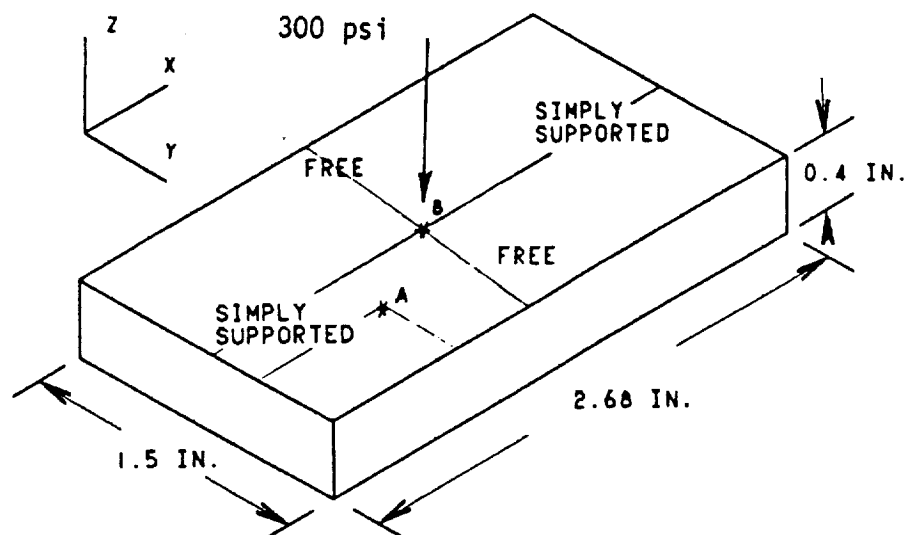


Figure A2.1: Illustration of simply-supported plate test.

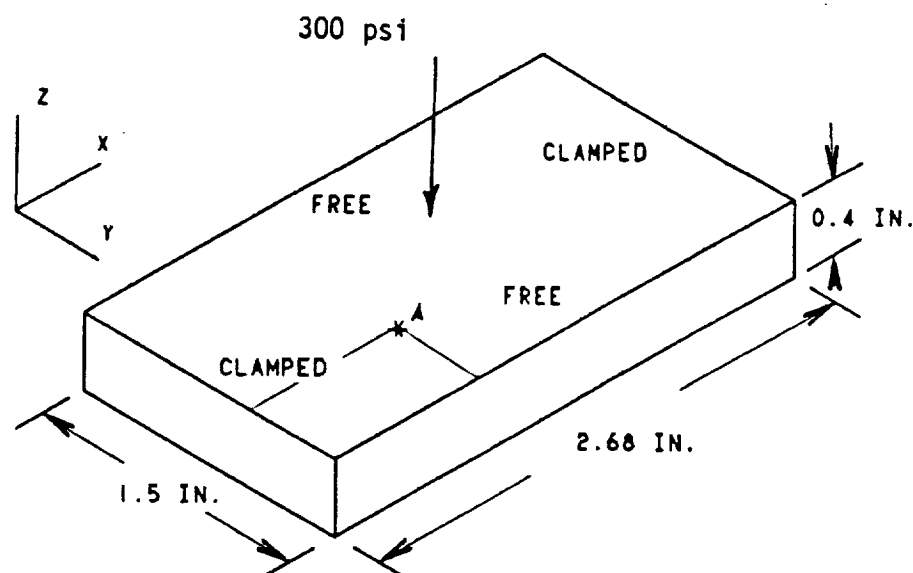


Figure A2.2: Illustration of clamped plate test.

ITEM	LOCATION	TEST	STRESS-X	STRESS-Y
(3X3) 4-NODE	A	SIMPLY-SUP.	8868	1422
(3X3) 8-NODE	A	SIMPLY-SUP.	9708	806.7
(6X6) 4-NODE	A	SIMPLY-SUP.	8904	1094
(6X6) 8-NODE	A	SIMPLY-SUP.	9121	668.8
(6X3) 8-NODE	A	SIMPLY-SUP.	9137	689.1
(12X6) 8-NODE	A	SIMPLY-SUP.	8983	655.8
(12X12) 8-NODE	A	SIMPLY-SUP.	8986	642.6
EXACT	A	SIMPLY-SUP.	8949	970.7
(4X4) 8-NODE	B	SIMPLY-SUP.	10400	917
(12X12) 8-NODE	B	SIMPLY-SUP.	10070	831.7
(3X3) 8-NODE	A	CLAMPED	2926	507.6
(6X6) 8-NODE	A	CLAMPED	2368	343
(3X3) 8-NODE*	A	CLAMPED	256	30
(6X6) 8-NODE*	A	CLAMPED	1242	167
(12X12) 8-NODE*	A	CLAMPED	1870	248
(3X3) 16-NODE	A	CLAMPED	2926	507
(6X6) 16-NODE	A	CLAMPED	2368	343
(12X12) 16-NODE*	A	CLAMPED	2226	295

STRESSES GIVEN IN PSI

\* THICK SHELL ELEMENT

Table 6: Results from plate test.

THICKNESS	EXACT		(12X12) 8-NODE	
	STRESS-X	STRESS-Y	STRESS-X	STRESS-Y
0.4 IN	8949	970	8986	642
0.2 IN	35800	3882	35820	3193
0.1 IN	143200	15530	143100	14110

STRESSES GIVEN IN PSI

Table 7: Results from plate test (thickness reduced).

ORIGINAL PAGE IS  
OF POOR QUALITY

## A2.10 SOLID ELEMENT TESTS

SUPERB's solid elements were also subjected to simple tests. The tests were to determine the element's ability to model both uniaxial and bending forces. In order to accomplish this a beam, shown in figure A2.3, and a column, shown in figure A2.4, were subjected to specific loadings. It was believed the beam test would induce the necessary bending action where as the column test would test the element's ability to model uniaxial loading. As expected, the solid elements modeled the beam test very poorly. Only by increasing the number of elements through the depth of the beam could the actual stress field be predicted. However, the solid elements did model the uniaxial loading quite well but this was not suprising. The results of both tests are given in tables 8 and 9.

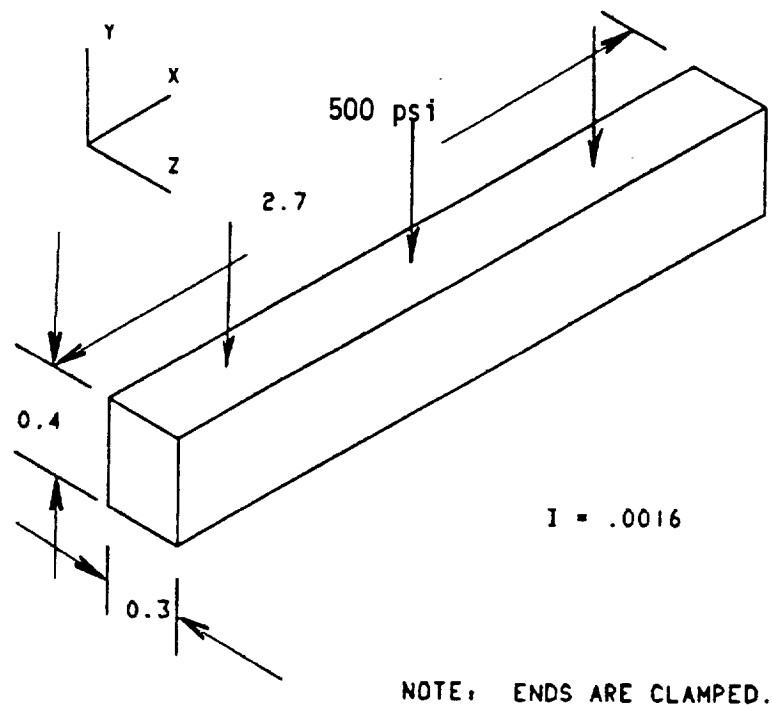


Figure A2.3: Illustration of beam test.

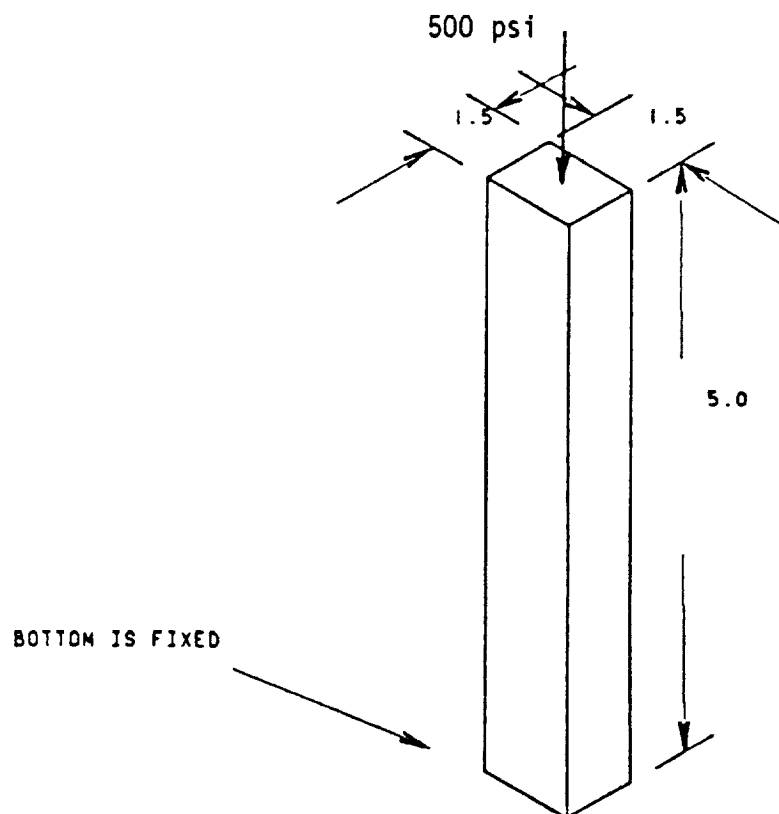


Figure A2.4: Illustration of column test.

#### FIBER STRESS IN BEAM TEST

ITEM	BEAM RESULTS	EXACT
(4X1X1) 8-NODE	-1851	-5695
(4X2X1) 8-NODE	-1774	-5695
(4X1X1) 20-NODE	-5800	-5695
(4X2X1) 20-NODE	-5900	-5695

STRESSES GIVEN PSI

Table 8: Results from beam test.

ITEM	COLUMN TEST	EXACT
(4X1X1) 8-NODE	-500 PSI	-500 PSI
(4X1X1) 20-NODE	-500 PSI	-500 PSI

STRESSES GIVEN IN PSI

Table 9: Results from column test.

### APPENDIX 3 EXACT SOLUTION FOR BENCHMARK TESTS

In reference (10), a series solution for a flat plate simply-supported on two opposing sides and free on the remaining two was found. By applying this solution to the benchmark plate geometry, accuracy of the plate elements were checked. It must be noted, though, that this solution does not include shear deformation effects.

So,

$$w = \left[ \frac{4pa^4}{\pi^5 D} \right] \sum_{m=1,2,3}^{\infty} \left[ \frac{1}{m^5} \right] \sin \left[ \frac{m\pi x}{a} \right] + \left[ \frac{pa^4}{D} \right] \sum_{m=1,2,3}^{\infty} \left[ A_m \cosh \left[ \frac{m\pi y}{a} \right] + \right.$$

$$\text{Where,} \quad B_m \left[ \frac{m\pi y}{a} \right] \sinh \left[ \frac{m\pi y}{a} \right] \sin \left[ \frac{m\pi x}{a} \right] \quad , \quad D = \left[ \frac{Et^3}{12(1-\nu^2)} \right]$$

$$A_m = \left[ \frac{4}{m^5 \pi^5} \right] \left[ \frac{\nu(1+\nu) \sinh \alpha_m - \nu(1-\nu) \alpha_m \cosh \alpha_m}{(3+\nu)(1-\nu) \sinh \alpha_m \cosh \alpha_m - (1-\nu)^2 \alpha_m^2} \right]$$

$$B_m = \left[ \frac{4}{m^5 \pi^5} \right] \left[ \frac{\nu(1-\nu) \sinh \alpha_m}{(3+\nu)(1-\nu) \sinh \alpha_m \cosh \alpha_m - (1-\nu)^2 \alpha_m^2} \right] \quad , \quad \alpha_m = \left[ \frac{m\pi b}{2a} \right]$$

Where  $w$  is the displacement normal to the plate's surface.  
Then,

$$\sigma_x = - \left[ \frac{Ez}{1-\nu^2} \right] \left[ \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right]$$

$$\sigma_y = - \left[ \frac{Ez}{1-\nu^2} \right] \left[ \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right]$$

$$\tau_{xy} = - \left[ \frac{Ez}{1+\nu} \right] \left[ \frac{\partial^2 w}{\partial x \partial y} \right]$$

Thus, by implementing the necessary parameters,

$a = 2.68$  in.       $b = 1.5$  in.

$E = 30 \times 10$  psi       $\nu = 0.3$

$p = 300$  psi

the solution was obtained.

APPENDIX 4  
NUMERICAL ROUTINES USED  
IN  
FEM CONSTRUCTION

Two routines were required to aid in construction of the FEM. The first was necessary to place points on the trochoidal bore at a specific angle relative to the outer shell's center of radius. This requirement is shown in Figure A4.1.

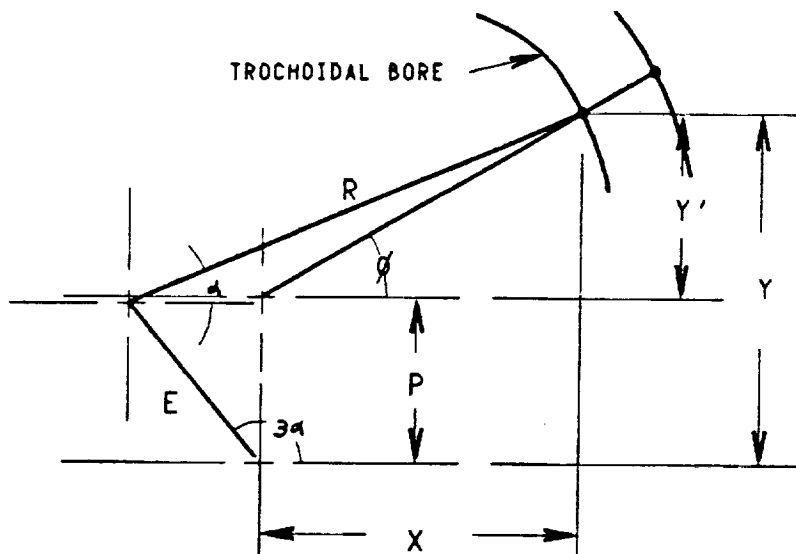


Figure A4.1: Illustration of the construction point placement.

In order to accomplish this a iteration routine was devised to solve equation 4.1:

$$\text{Eq. 4.1: } \alpha = \sin^{-1} \left[ \frac{(e \cos 3\alpha + R \cos \alpha) \tan \phi + P - e \sin 3\alpha}{R} \right]$$

A second routine was needed to find the X coordinate of the trochoidal bore given any Y coordinate. Thus, the second routine solved Equation 4.2:

$$\text{Eq. 4.2: } \alpha = \sin^{-1} \left[ \left[ \frac{1}{R} \right] [Y - e \sin(3\alpha)] \right]$$



## REFERENCES

- 1) Jones, C., "Advanced Development of Rotary Stratified Charge 750 and 1500 HP Military Multi-fuel Engines at Curtis-Wright," SAE Paper 840460, presented at the International Congress and Exposition, Detroit, Feb. 1984.
- 2) Matsuura, K., et. al., "The Behavior of a Rotary Engine Apex Seal Against the Trochoidal Surface," Bull. of JSME, Vol. 21, No. 161, Nov. 1978.
- 3) Eberle, M. K., and Klomp, E. D., "An Evaluation of the Potential Performance Gain from Leakage Reduction in Rotary Engines," SAE Paper 730117, presented at the International Automotive Engineering Congress, Detroit, Jan. 1973.
- 4) Behling, R. D., and Weise, E., "Lubricant and Fuel Requirements and General Performance Data of Wankel Rotary Piston Engines," SAE Paper 730048, presented at the International Automotive Engineering Congress, Detroit, Jan. 1973.
- 5) Rodgers, T. W., et. al., "Lubricant Studies in Rotary Combustion Engines," SAE Paper 720467, presented at the National Automotive Engineering Meeting, Detroit, May 1972.
- 6) Prasse, H. F., McCormac, H. E., and Anderson, R. D., "A Critical Analysis of the Rotary Engine Sealing Problem," SAE Paper 730118, presented at the National Automobile Engineering Meeting, Detroit, May 1972.
- 7) Matsuura, K., et. al., "The Relative Behavior of a Rotary Engine Apex Seal to the Walls of a Slot," Bull. of the JSME, Vol. 19, No. 137, Nov. 1976.
- 8) Ansdales, R. F., The Wankel RC Engine, A. S. Barnes & Co., 1969.
- 9) "SUPERB Users Manual," Structural Dynamics Research Corporation, 1983.
- 10) Timoshenko, S., and Woinowsky-Krieger, S., Theory of Plates and Shells, McGraw-Hill Book Company, 1968.

- 11) "SUPERTAB Users Manual," Structural Dynamics Research Corporation, 1983.
- 12) Jones, C., "Development of the Rotary Engine," SME Technical Paper MM73-650, presented at the Rotary Engine Symposium, Detroit, Nov. 1973.
- 13) Meng, P. R., Hady, W. F., Barrows, R. F., "An Overview of the NASA Rotary Engine Research Program," SAE Paper 841018, presented at the West Coast International Meeting & Exposition, San Diego, Aug. 1984.
- 14) Norbye, J. P., The Wankel Engine Design Development Applications, Chilton Book Company, 1971.
- 15) Knoll, J., "A Dynamic Analysis of Rotary Combustion Engine Seals," a thesis presented to Michigan Technological University, Nov. 1983.
- 16) Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, 1981.
- 17) Vinson, J. R., Structural Mechanics: The Behavior of Plates and Shells, John Wiley & Sons, 1974.
- 18) Ugural, A. C., and Fenster, S. K., Advanced Strength and Applied Elasticity, Elsevier, 1982.

# **Appendix**

## **B**

**DEVELOPMENT OF A PREPROCESSOR THAT GENERATES  
FINITE ELEMENT MODELS OF ROTARY COMBUSTION ENGINE  
CENTER HOUSINGS**

by

**WILLIAM M. Lychuk**

**A THESIS**

submitted in partial fulfillment of the requirements

for the degree of

**MASTER OF SCIENCE IN MECHANICAL ENGINEERING**

**MICHIGAN TECHNOLOGICAL UNIVERSITY**

**1985**

**LIBRARY**  
**MICHIGAN TECHNOLOGICAL UNIVERSITY**  
**HOUGHTON, MICHIGAN**

This thesis, " Development of a Preprocessor that Generates Finite Element Models of Rotary Combustion Engine Center Housings, " is hereby approved in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN MECHANICAL ENGINEERING.

DEPARTMENT: Mechanical Engineering -

Engineering Mechanics

*Carl Gilmann*

Thesis Advisor

*Harold V. Lark*

Head of Department

*1-28-86*

Date

231737

NOV 6 '86

## **Abstract**

This thesis documents the development of a specialized preprocessor that will generate a finite element model of different rotary combustion engine center housing geometries. The specialized preprocessor has been written to be used in conjunction with General Electric's Computer Aided Engineering software, specifically, IDEAS. The only user-supplied inputs required by the specialized preprocessor are easily measured parameters that describe the center housing geometry.

When executed, the FORTRAN coded specialized preprocessor creates two files - a universal file and a program file. The universal file contains data in universal format that describes the housing geometry. Universal format is a standard that has been defined by General Electric's Computer Aided Engineering. The program file contains commands that are understood by the programmability allowed within IDEAS. The commands guide the algorithms in IDEAS through the generation of the meshes, nodes, and elements.

## Acknowledgments

The author would like to express his thanks to Dr. Carl Vilmann and Dr. Chris Passerello for their guidance, encouragement and friendship throughout this project. Both have helped me to maintain my perspective and to understand what was to be accomplished.

Thanks is expressed to Dr. Thomas Grimm and Dr. William Bulleit for reviewing this paper and serving on the examination committee.

Love and affection to my fiance Katherine Drakos for her unending love and patience.

Most importantly, the author wishes to express his deepest love to his parents, to whom, he owes everything that he has achieved.

# Table of Contents

Abstract .....	i
Acknowledgments .....	ii
List of Illustrations .....	vi
Glossary of Abbreviations .....	vii
CHAPTER 1: Introduction .....	1
1.1 Rotary Combustion Engine Center Housing Geometry .....	3
1.2 Rotary Combustion Engine Development .....	7
1.2.1 Major Sources of Problems in the Development of the RCE .....	8
1.2.2 Modifications of the RCE .....	9
1.3 Finite Element Analysis Background .....	11
1.4 Development of Automatic Finite Element Model Generators .....	13
1.4.1 Definition of the Specialized Preprocessor .....	14
CHAPTER 2: Universal File Development .....	16
2.1 Geometric Entity Definition .....	16
2.1.1 Rib Definition .....	17
2.1.2 Channel Definition .....	20
2.1.3 Bore Insert Definition .....	21
2.1.4 Intake Port Definition .....	22



2.1.5 Exhaust Port Definition .....	23
2.1.6 Spark Plug Port Definition .....	23
2.2 Geometric Parameter Definition .....	23
2.2.1 Rib and Channel Input Parameter Definition .....	30
2.2.2 Intake and Exhaust Port Input Parameter Definition .....	31
2.2.3 Spark Plug Port Input Parameter Definition .....	32
2.3 Numbering Patterns of the Geometric Entities .....	33
2.4 Reduction & Implementation of Input for Universal File Generation .....	36
2.4.1 Housing Dimension Input Reduction and Implementation .....	36
2.4.2 Cylindrical Region Input Reduction and Implementation .....	38
2.4.3 Rectangular Region Input Reduction and Implementation .....	42
2.4.4 Intake Port Input Reduction and Implementation .....	44
2.4.5 Exhaust Port Input Reduction and Implementation .....	47
2.4.6 Spark Plug Port Input Reduction and Implementation .....	48
<b>CHAPTER 3: Program File Development .....</b>	<b>49</b>
3.1 Mesh Definition .....	49
3.2 Node and Element Definition .....	50
<b>CHAPTER 4: User's Guide .....</b>	<b>51</b>
4.1 Obtaining the Input .....	51
4.2 Input Data File .....	54
<b>CHAPTER 5: Conclusions .....</b>	<b>55</b>
<b>References .....</b>	<b>59</b>
<b>Appendix 1: Example of a Universally Formatted Data File .....</b>	<b>61</b>
<b>Appendix 2: Example of an Input Data File .....</b>	<b>65</b>

<b>Appendix 3: Examples of Numbering Patterns</b> .....	<b>68</b>
APPENDIX 3.1: Numbering Pattern of the Splines .....	69
APPENDIX 3.2: Numbering Pattern of the Edges .....	71
APPENDIX 3.3: Numbering Pattern of the Surfaces .....	73
APPENDIX 3.4: Numbering Pattern of the Volumes .....	75

## List of Illustrations

Figure 1. Rotary Combustion Engine .....	1
Figure 2. Major Components of the Rotary Combustion Engine .....	2
Figure 3. The execution of the Otto cycle in the RCE .....	4
Figure 4. Generation of the Epitrochoid .....	5
Figure 5. Rotor with Seals .....	6
Figure 6. Generation of the Trochoid .....	7
Figure 7. Crowning of the Apex Seal .....	9
Figure 8. Spark Plug Spacing and Shapes of the Combustion Chamber ....	10
Figure 9. Stratified Charging of a Rotary Combustion Engine .....	10
Figure 10. Fundamental Components of the Center Housing .....	18
Figure 11. Ribs, Channels and Ports .....	19
Figure 12. Modeling of the Two Rib Types .....	20
Figure 13. Modeling of the Two Channel Types .....	21
Figure 14. Example of Combined Rib and Channel Types .....	22
Figure 15. Modeling of the Intake Port .....	24
Figure 16. Modeling of the Exhaust Port .....	25
Figure 17. Modeling of the Spark Plug Port .....	26
Figure 18. Typical Drawing of the Center Housing .....	27
Figure 19. Illustration of the Different Regions in the Center Housing ....	29

Figure 20. Illustration of the Intake and Exhaust Port Input Data .....	32
Figure 21. Example of a Numbering Pattern .....	34
Figure 22. Pictures Generated from Numbering Pattern .....	35
Figure 23. Illustration of the Three Rib/Channel Segment Types .....	37
Figure 24. Determination of the Center of the Cylindrical Region .....	39
Figure 25. Graph of Alpha versus Phi .....	41
Figure 26. Illustration of the Seven Phi Angles .....	43
Figure 27. Illustration of the Seven Y-Coordinates .....	45
Figure 28. Critical Area for Distorted Elements .....	47
Figure 29. Angles Defining Radius of the Intake or Exhaust Port .....	54
Figure 30. FEM Generated by the SP .....	56
Figure 31. A second FEM Generated by the SP .....	57
Figure 32. Example Model Generated by the Universal File .....	64
Figure 33. Numbering Pattern of the Splines .....	69
Figure 34. Illustration of the Numbering Pattern of the Splines .....	70
Figure 35. Numbering Pattern of the Edges .....	71
Figure 36. Illustration of the Numbering Pattern of the Edges .....	72
Figure 37. Numbering Pattern of the Surfaces .....	73
Figure 38. Illustration of the Numbering Pattern of the Surfaces .....	74
Figure 39. Numbering Pattern of the Volumes .....	75
Figure 40. Illustration of the Numbering Pattern of the Volumes .....	76

## **Glossary of Abbreviations**

<i><b>Abbreviation</b></i>	<i><b>Description</b></i>
<b>AAD</b>	Advanced Adiabatic Diesel
<b>BMEP</b>	Brake Mean Effect Pressure
<b>FEM</b>	Finite Element Model
<b>GE/CAE</b>	General Electric / Computer Aided Engineering
<b>MPG</b>	Miles Per Gallon
<b>PF</b>	Program File
<b>RCE</b>	Rotary Combustion Engine
<b>RPE</b>	Reciprocating Piston Engine
<b>SP</b>	Specialized Preprocessor
<b>UF</b>	Universal File

## CHAPTER 1: Introduction

The Rotary Combustion Engine (RCE) was first introduced in the early 1950's by Felix Wankel, a German inventor (Figure 1).

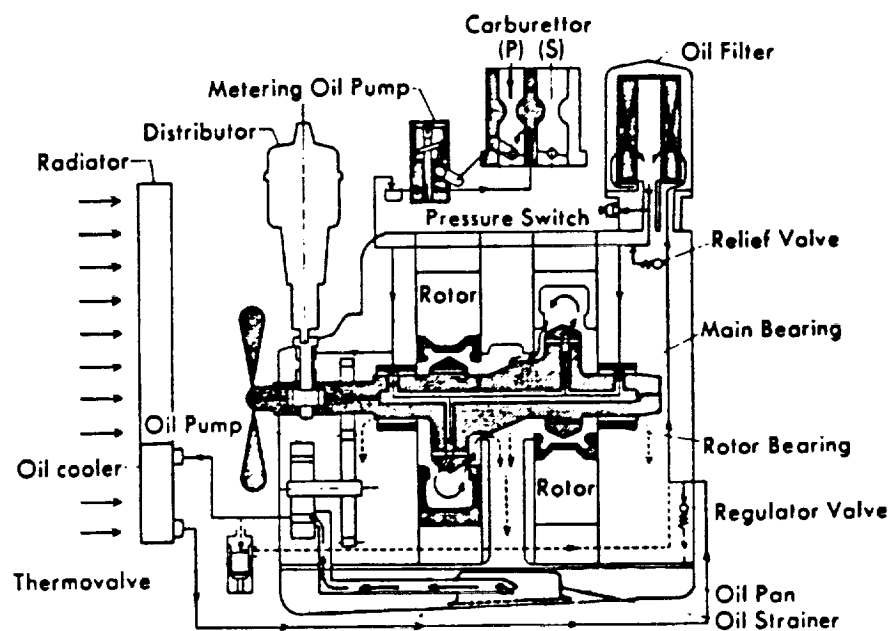


Figure 1. Rotary Combustion Engine

The RCE is an internal combustion engine which consists of three major components - rotor, center housing and end plates. The end plates are bolted to each side of the center housing. The triangularly shaped rotor is located in the trochoidally shaped chamber of the center housing (Figure 2).

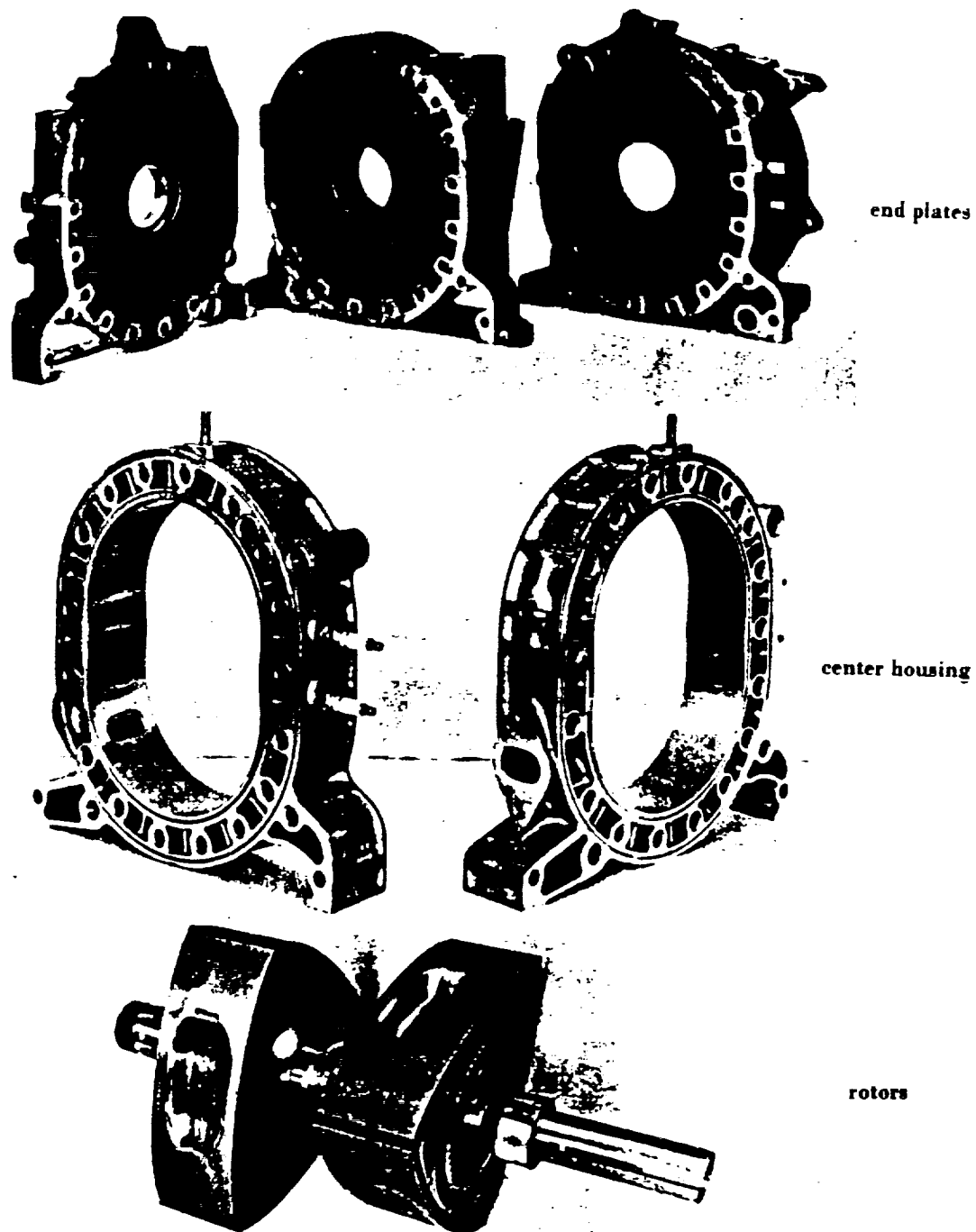


Figure 2. Major Components of the Rotary Combustion Engine

The rotation of the rotor causes different sized chambers to be created between each face of the rotor and the trochoidal surface. It is in these chambers that the different phases of the Otto cycle - intake, compression, combustion and exhaust - are executed (Figure 3). One Otto cycle is executed by each face of the rotor every revolution (i.e. three complete Otto cycle are executed per rotor revolution).

### 1.1 Rotary Combustion Engine Center Housing Geometry

A prominent feature of the center housing geometry is the trochoidal shape of the inner surface. The generation of this shape is accomplished by first constructing an epitrochoid. By definition, an epitrochoid is the locus of points created by a point on the radius of a circle which rolls without slip around a base circle. The familiar two-lobed epitrochoid is obtained when the base circle is equal to twice the rolling circle (Figure 4). The dimensions needed to describe the epitrochoid are the base circle radius, the rolling circle radius (one half of the base circle radius) and the distance of the generating point from the rolling circle's center, the eccentricity.

The epitrochoidally shaped inner bore of the center housing is constructed so that the rotor, when inserted, will create line contact at all three apexes. To enhance the sealing capacity between the rotor apexes and epitrochoidal surface, seals are placed at the apex of each rotor (Figure 5). To account for the extra space required by the seals, the epitrochoidal surface has to be expanded. The expanded curve is called a trochoid. During operation, the motion of the center of the radius of the apex seal traces an epitrochoidal curve and the motion of the apex seal tip traces a trochoidal curve.

The trochoidal shape is derived from expanding the epitrochoid a perpendicular distance equal to the radius of the apex seal. A family of curves can be created by specifying different magnitudes of the perpendicular distance (Figure 6).



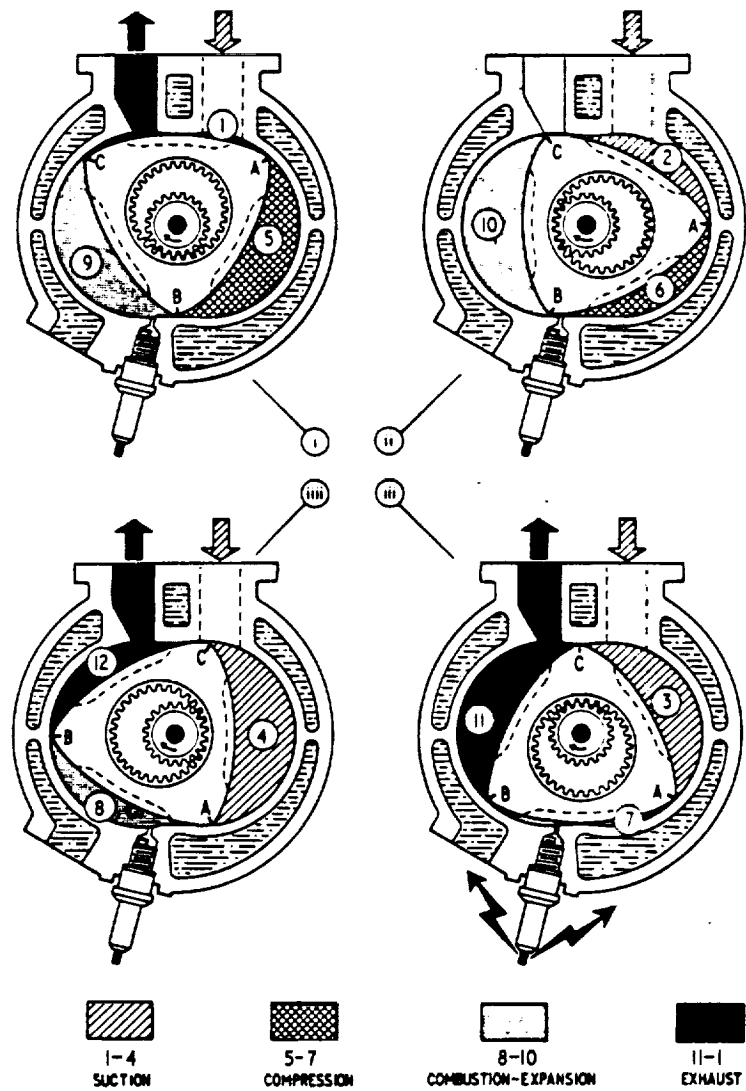


Fig. 5.3. 1-4 induction  
5-7 compression  
8-10 expansion  
11-1 exhaust

Figure 3. The execution of the Otto cycle in the RCE

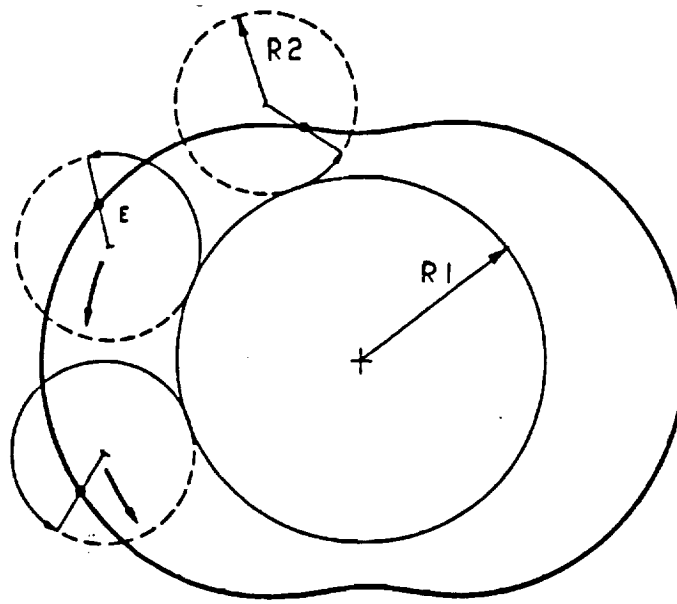


Figure 4. Generation of the Epitrochoid

Ansdale (1) gives the rectangular coordinates of any point on the trochoidal surface as:

$$x = E \times \cos(3\alpha) + R \times \cos(\alpha) + A \times \cos(\alpha + \theta) \quad (1.1a)$$

$$y = E \times \sin(3\alpha) + R \times \sin(\alpha) + A \times \sin(\alpha + \theta) \quad (1.1b)$$

where E is the eccentricity, R is the radius of the generating circle, A is the perpendicular distance between the epitrochoid and trochoid, and theta is the angle of obliquity. Theta is defined by:

$$\theta = \cos^{-1} \left[ \frac{(R + 3 \times \cos(2 \times \alpha))}{\sqrt{9 \times E^2 + R^2 + 6 \times E \times R \times \cos(2 \times \alpha)}} \right] \quad (1.1c)$$

Theta is a measure of the angle, relative to the normal of the trochoidal curve, that the apex seal rotates through during operation. Theta is a minimum at the major and minor axes and reaches a maximum value midway between the two axes.

## 1.2 Rotary Combustion Engine Development

After Felix Wankel persuaded an obscure German motorcycle and small car manufacturer, NSU, to help develop the RCE, many other companies, including General Motors, Porsche, Rolls Royce, Curtis-Wright and Toyo Kogyo, have attempted to make the

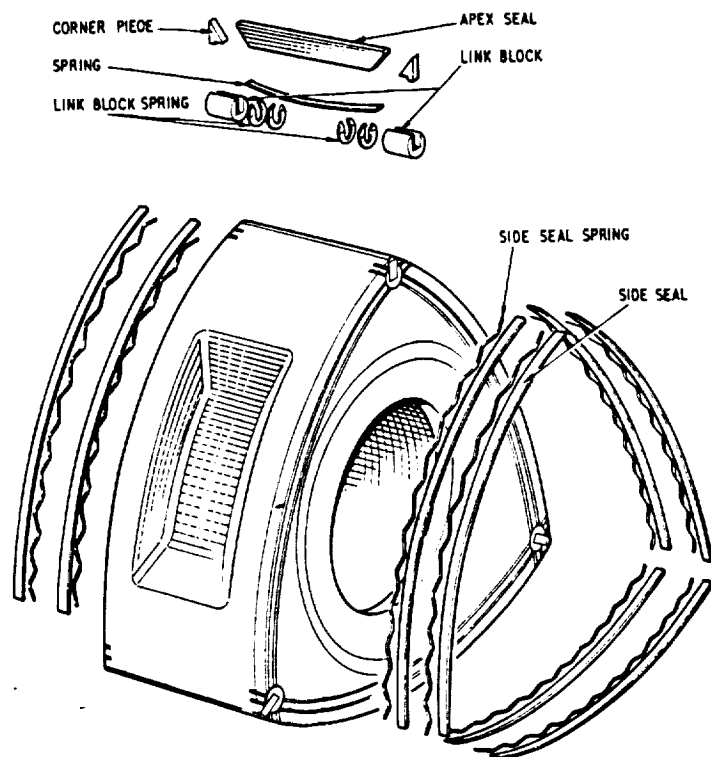


Figure 5. Rotor with Seals

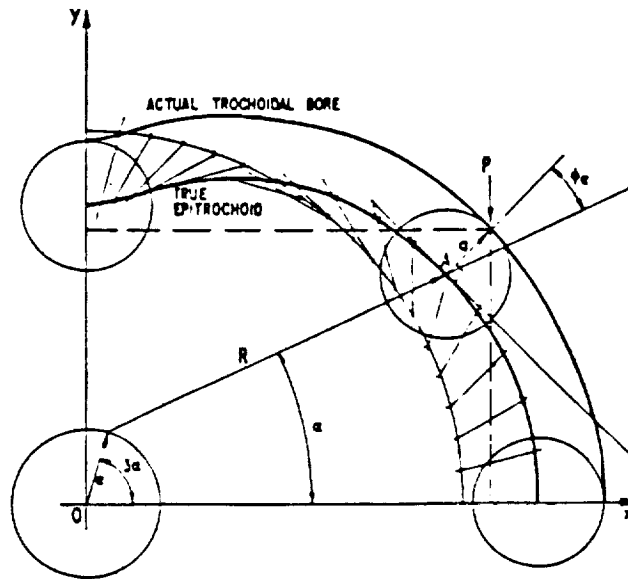


Figure 6. Generation of the Trochoid

RCE's actual performance characteristics approach its theoretical possibilities. To date in the automotive industry, Toyo Kogyo's Mazda RX7 is the only production automobile still using the RCE.

During its initial development, the RCE was hailed as the engine of the future that would replace the reciprocating piston engine (RPE). In many areas, the RCE has better performance characteristics than its RPE counterpart. For RCE and RPE engines of comparable power output, the RCE has less than half the weight, is almost half the size, produces less noise and vibration, and uses less than half the moving parts (2). But the RCE is plagued with a 12% higher specific fuel consumption and higher exhaust emissions (3). These are two of the major problems that have prevented the RCE from living up

to its early expectations.

### **1.2.1 Major Sources of Problems in the Development of the RCE**

The major sources of the problems preventing the widespread acceptance of the RCE include:

1. When combustion pressures are high, imperfect sealing of the apex seal on the trochoidal surface will permit leakage into the leading and/or trailing chamber. The leakage, because it comes from the hydrocarbon-rich end of the combustion chamber, leads to higher hydrocarbon emissions (4).
2. Residuals are the fraction of the gasses remaining in the combustion chamber at the end of the exhaust cycle. The residuals mix with the incoming air fuel mixture diluting it largely with inert gasses. The residuals cause cycle-to-cycle variations in the combustion mixture (5).
3. At high operating speeds, the apex seal is subjected to a high contact force from the trochoidal surface. The friction forces, resulting from the contact force, cause excessive wearing of the apex seal. As the apex seals wear, more leakage is allowed into the leading and/or trailing chambers (6).
4. About forty percent of the combustion chamber surface is on the rotor. The balance is on the trochoid. A layer of unburned hydrocarbons is formed on these surfaces as the flame front is quenched by these relatively cool surfaces. These unburned hydrocarbons contribute to the high exhaust emissions (7).

### 1.2.2 Modifications of the RCE

In an effort to enhance the operating conditions of the RCE many modifications have been proposed. Some of the significant modifications include:

1. In order to improve the apex seal conformability to the trochoidal surface, crowning of the apex seal in the radial direction was applied (8). Although the crowning is only ten to thirty microns, its effect on the brake mean effect pressure (BMEP) is an increase of two to eight percent (Figure 7).

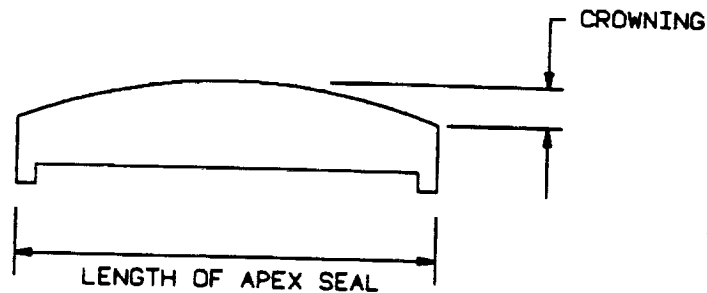


Figure 7. Crowning of the Apex Seal: The crowning has been amplified for illustrative purposes.

2. The shape of the combustion chamber recess and the spark plug location influences the amount of residuals (9). By increasing the distance between the two spark plugs and altering the combustion chamber recess, the BMEP is increased and the brake specific fuel consumption is reduced (Figure 8).
3. The shape and opening and closing timings of the port, especially the intake port, can affect the volumetric efficiency by as much as twenty percent (10).
4. Stratified charging of the RCE has increased the miles per gallon (MPG) rating by as much as fifty percent (11). (Figure 9)

Currently, there is an effort to develop advanced adiabatic diesel (AAD) engines. An AAD engine uses ceramic coatings on critical engine components to allow for higher operating temperatures. These higher operating temperatures significantly increase the

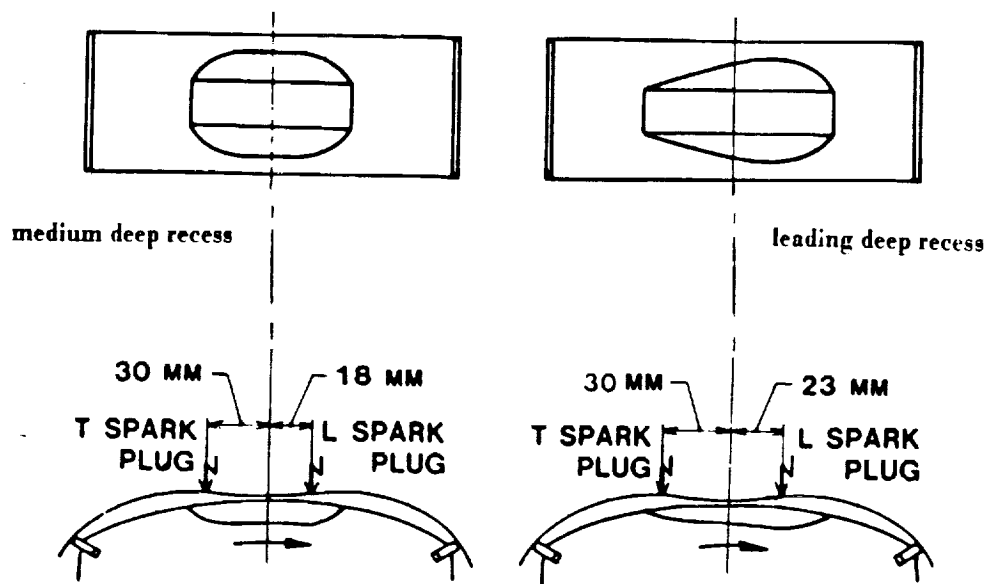


Figure 8. Spark Plug Spacing and Shapes of the Combustion Chamber

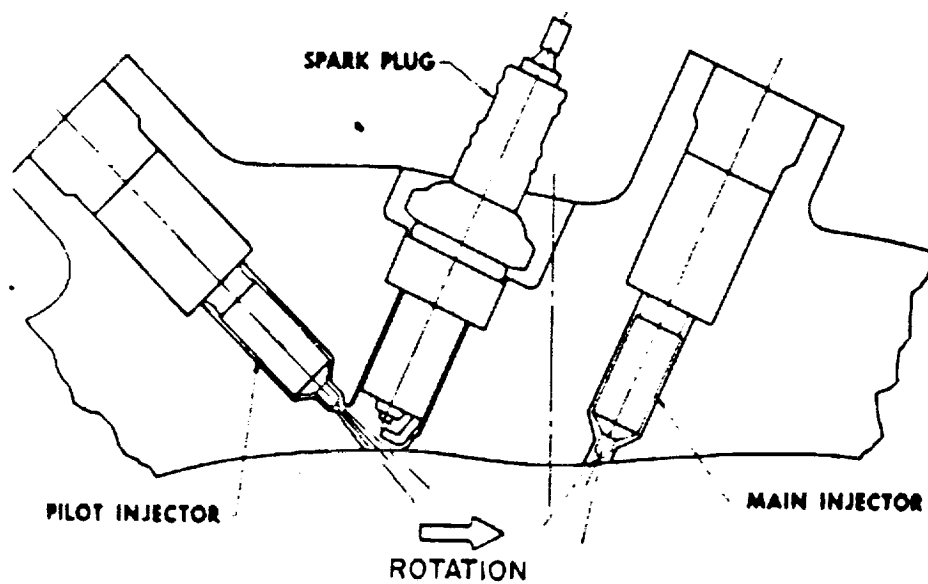


Figure 9. Stratified Charging of a Rotary Combustion Engine

engine's MPG rating and reduce certain exhaust emissions. The RCE is well suited for conversion to an AAD engine. Its multifuel tolerance allows it to burn diesel fuel and its fewer moving parts and simpler configuration make the application of a ceramic coating on the necessary components of the RCE easier than on the RPE.

### 1.3 Finite Element Analysis Background

Whenever any modifications or developments of the RCE are proposed, both the efficiency derived from these changes and their effect on the structural integrity of the engine must be evaluated. Typically, these types of evaluations have been made by construction and testing of engine prototypes. Since this construction and testing can become very costly, accurate analytical modeling of the proposed modifications can save both time and money. Instead of building a prototype engine out of metal, a mathematical model of the engine can be developed using finite elements.

The Finite Element Method (FEM) is a numerical procedure used for solving differential equations. The method involves dividing a physical continuum into a finite number of geometric units, the finite elements. The equations of the properties of the individual elements can be assembled using different approaches - direct, variational, weighted residuals, or energy balance. By solving these equations simultaneously using Gaussian Elimination or a similar method, an approximation of the exact solution can be obtained.

The solution of a general continuum problem by the finite element method follows an orderly step-by-step process. The first step of a finite element analysis is to determine how accurately the geometry needs to be modeled in order to obtain stresses and displacements that closely approximate the actual values. This determination is based primarily on preliminary analysis, previous experience and a physical understanding of the



problem at hand. For the center housing, the most significant geometry that was not modeled was the bolt holes in the ribs. Bolts are passed through the holes and are used to hold the two end plates to the housing. It was determined in a previous study that the exclusion of the bolt holes from the finite element model did not significantly affect the stress and deformation results (11). Other noncritical geometry, such as the grooves for the gaskets, was not modeled because its effect on the results would be insignificant.

After the geometry is created, it is discretized into subdivisions or finite elements. Before the elements can be defined, nodes and meshes are described. Meshes define the boundaries within which the nodes and elements are located. The nodes, specified by spatial coordinates, define the elements. Loading and restraint conditions are applied at the nodes. Element parameters - number, size and type - significantly affect the accuracy of the stress and deformation results. The element parameters that were used to create the FEM of the housing were validated in a previous study (11).

The final steps in the finite element analysis of the center housing include assuming a displacement model, and from that model, deriving the individual element stiffness matrices. These individual matrices are assembled and then modified to account for the restraint conditions. The matrices were formulated according to the equation:

$$\{F\} = [K]\{X\} \quad (1.3a)$$

where  $\{F\}$  is the column matrix of nodal loads,  $[K]$  is the combined stiffness matrix and  $\{X\}$  is the matrix of nodal displacements. When equation 1.3a was solved,  $\{X\}$  being the unknown, the nodal displacements are known. The stresses are derived from the displacements by using the appropriate solid mechanics equations.

#### 1.4 Development of Automatic Finite Element Model Generators

Due to the complex nature of the RCE center housing geometry, the amount of data necessary to create a FEM is enormous. To aid the user in the construction of the FEM, data preprocessors, such as IDEAS, have been written. Even with the aid of these powerful general purpose preprocessors, it can take hundreds of hours to build one model (12). This time intensive nature of the FEM creation greatly adds to the cost of the FEM design procedure. This can prohibit the analysis of many alternate designs.

Automated design and optimization programs have been developed to reduce the time-intensive nature of finite element model creation. Typically, the programs consist of two major components - the controller and the analyzer. The analyzer performs two functions. The first function is to accept the user supplied design variables - maximum allowable stresses or critical dimensions - that are not altered during the optimization. The second function of the analyzer is to calculate the stresses and deflections of the structure. The optimizer provides input to the analyzer, and based on the results of the analyzer, determines what changes should be made to reduce the mass of the geometry being optimized.

The major drawbacks of the optimization programs include the requirement on the user to create the initial model and the limitations of the type of geometry applicable for optimization. In general, geometries that can be stamped, rather than cast, are better suited for automated design. Bennett (13) cites several reasons for this. They include: problems maintaining an adequate finite element mesh on boundaries that are varied during the optimization, defining general shapes using a reasonable number of design variables, and imposing the proper constraints so that a realistic design results. Because of these restrictions, the automated design and optimization programs are best suited for only relatively complex structures that can be modeled with beams, thin shells or a combination of the two element types. Currently, the programs are used in the optimization

of automobile and aircraft bodies.

#### 1.4.1 Definition of the Specialized Preprocessor

There is a need to develop a program that can automatically create models of complex three dimensional geometry with a minimum of user supplied inputs. The current automated design and optimization programs are insufficient in that they not only require the user to create the initial model, but also prohibit the use of three-dimensional elements in the finite element model. This thesis outlines the development of a specialized preprocessor (SP) that will reduce the time necessary to create FEM's of different RCE center housing geometries. With a minimum of user supplied inputs, the SP will automatically generate a finite element model of the specified housing geometry. The automatic optimization of the model, as in the current programs, is beyond the scope of this thesis; it is left up to the user. The SP is designed only to build finite element models of different RCE center housing geometries. Being geared towards the building of a specific geometry, the specialized preprocessor requires only a minimum knowledge of the system by the user. With the automatic capability of the SP significantly reducing the time necessary to construct the FEM, alternate RCE center housing designs can be analyzed without time or cost becoming prohibitive.

The SP was developed to be used in conjunction with General Electric / Computer Aided Engineering's (GE/CAE) graphics package, specifically, IDEAS. The SP is a FORTRAN coded program. When executed, the SP creates two different files. The first file is formatted such that it can be read by the GE/CAE package. This file contains all the data necessary to describe the geometry of the RCE center housing. The file is in universal format; hence the name, Universal File (UF). The universal format is a standard that has been defined by GE/CAE (14-15). See Appendix 1 for an example of a universally formatted file. The second file created by the SP contains commands that can

be executed within IDEAS. The commands are necessary to guide the IDEAS subroutines through the generation of the meshes, nodes and elements on the geometry. The file uses the programming capabilities within IDEAS; hence the name Program File (PF).

There were several reasons why it was necessary to write two files - one UF and one PF - instead of just one UF or just one PF. The PF was necessary because certain entities that are needed to define a FEM, namely the meshes, can not be defined in the UF because GE/CAE did not include meshes when they defined this universal format. The PF contains commands that use the programmability allowed within IDEAS to guide the complicated algorithms in IDEAS through the generation of the meshes, nodes and elements. The UF is still necessary, although limited to defining geometry, because the programmability that is allowed within IDEAS is limited to two hundred variables - not sufficient to describe the complex geometry of the center housing, but powerful enough to guide the algorithms in IDEAS through the generation of the meshes, nodes and elements. In summary, the UF is needed to define the geometry, but it can not define the meshes, nodes and elements. The PF is needed to define the meshes, nodes and elements, but it can not be used to program the different geometric possibilities of the center housing.

## **CHAPTER 2: Universal File Development**

Before the development of the UF is described, the reader should have a good understanding of the universal format. For the RCE center housing, six geometric entities are required to completely describe the RCE center housing geometry. They are points, lines, splines, edges, surfaces, and volumes. GE/CAE has established a hierarchy that must be followed when describing these geometric entities in universal format. From lowest order to highest order entity, the hierarchy is:

- Points are used to define lines and splines.
- Lines and splines are used to define edges.
- Edges are used to define surfaces.
- Surfaces are used to define volumes.

As all these geometric entities are being created, they are assigned numeric labels. The universal format uses the labels of lower order entities to define upper order entities. As examples, two point labels could be used to define a line or six surface labels could be used to define a volume.

### **2.1 Geometric Entity Definition**

The first step required in the development of the UF was the decomposition of the complicated geometry of the center housing into smaller, simpler components. The most fundamental components are the inner and outer shells of the center housing and the space

between the shells (Figure 10). These shell components can be combined to form geometric building blocks that represents small but repeated sections of the center housing. These geometric building blocks are:

- Ribs.
- Channels.
- Intake port.
- Exhaust port.
- Spark plug ports.

By combining the building blocks in different combinations, the complicated center housing geometry can be formed. The ribs and channels are the primary building blocks. The majority of the center housing is formed by alternating ribs and channels. The intake and exhaust ports exist only once, if at all, in the center housing (in some engine configurations, the intake port is located in the end plates not in the center housing). The spark plug ports exist twice.

### **2.1.1 Rib Definition**

There are two different types of ribs - normal and recessed (Figure 12). A normal rib is defined as the inner and outer shell of the center housing and the material between the shells. A rib is designated as recessed when there is a gap between the material and the inner shell. This type of rib is used in areas where relatively high temperatures are expected - the spark plug region. The increased coolant flow allowed by the recess dissipates more heat in this area.

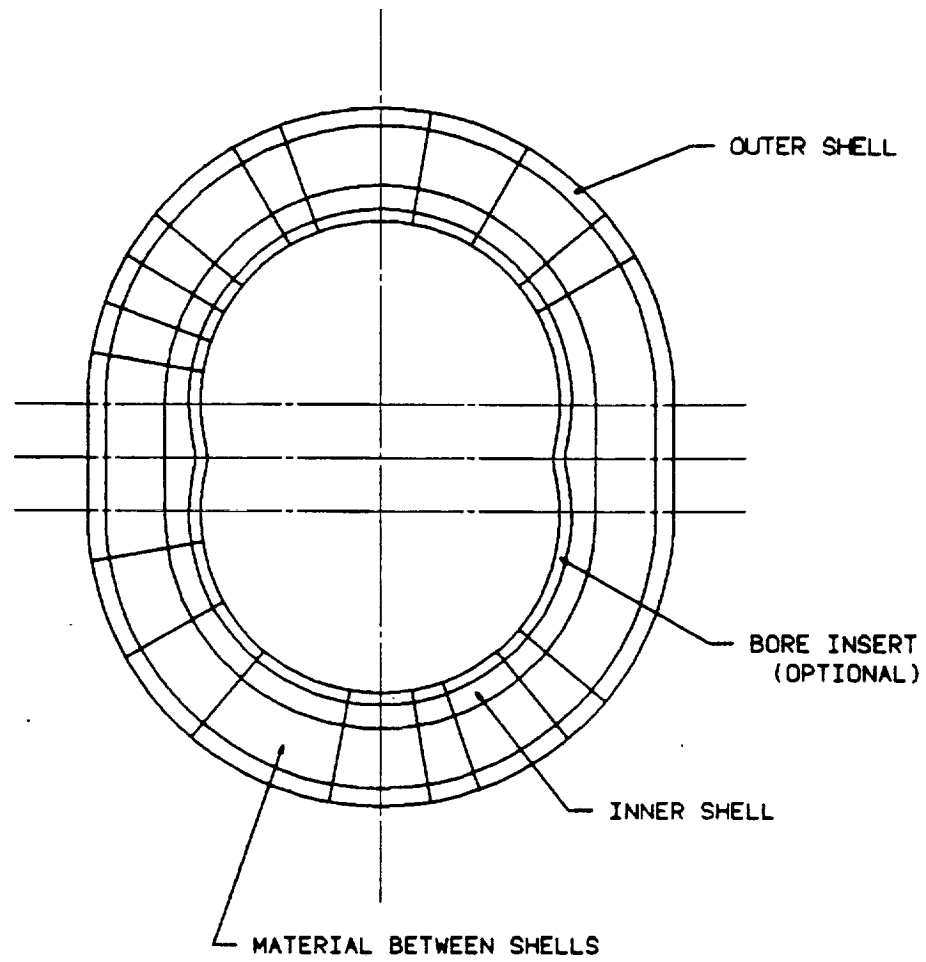


Figure 10. Fundamental Components of the Center Housing

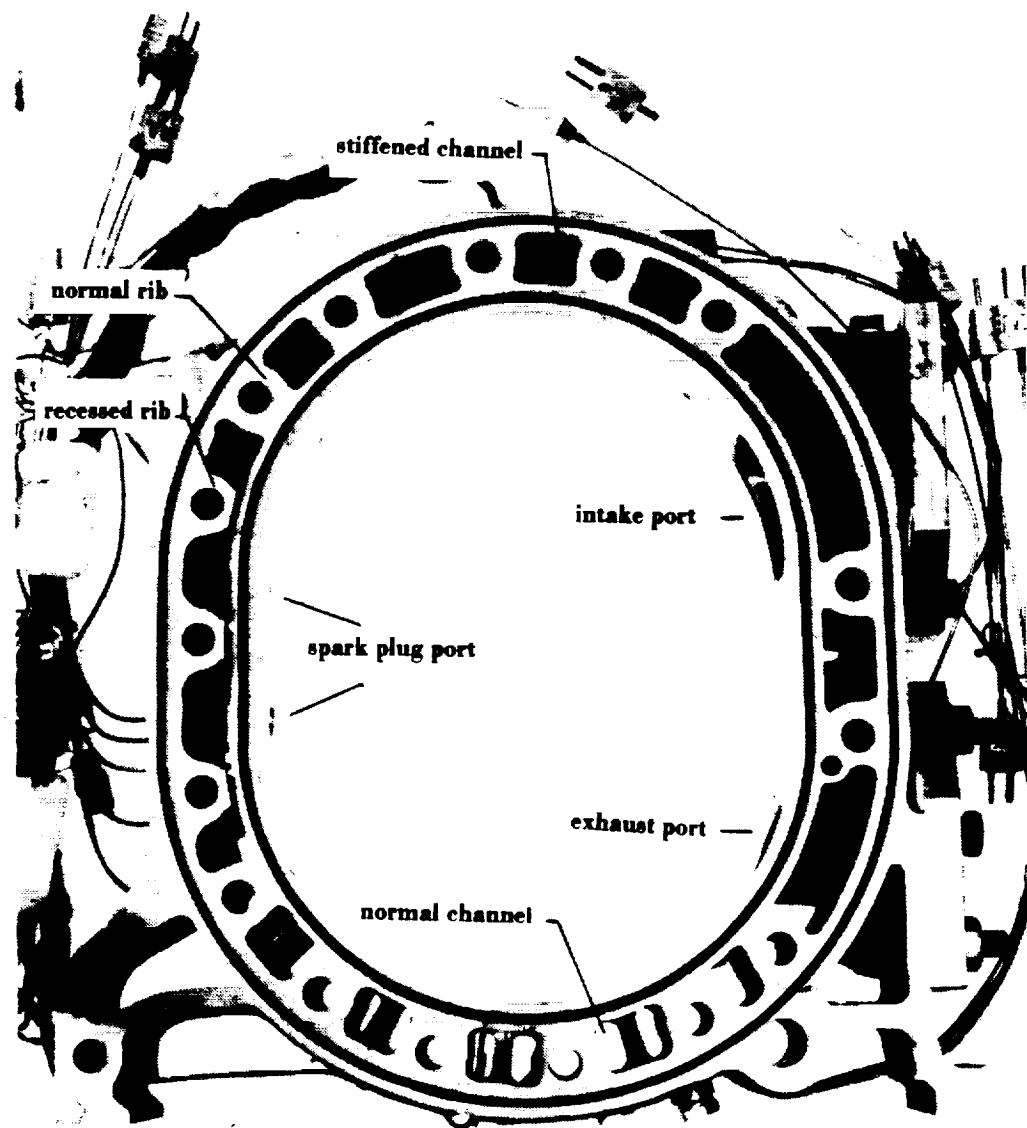
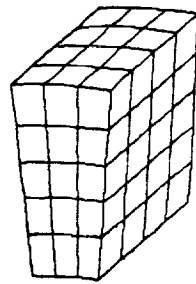
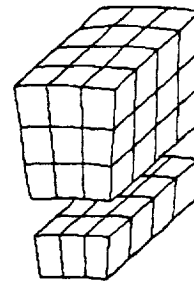


Figure 11. Ribs, Channels and Ports: as they appear in the Center Housing.





normal rib



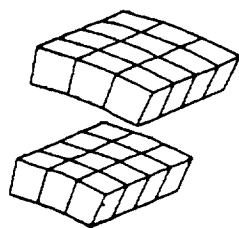
recessed rib

Figure 12. Modeling of the Two Rib Types

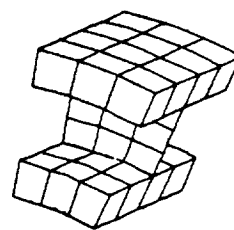
---

### 2.1.2 Channel Definition

There are two types of channels - normal and stiffened (Figure 13). A normal channel is defined as the inner and outer shell of the center housing. There is no material between the shells. Engine coolant flows through the vacancy between the shells. A channel is designated as stiffened when a thin plate at the midplane of the channel blocks the coolant flow. This type of channel is used in areas where relatively large deformations and low temperatures are expected - the compression region. The increased stiffness caused by the plate decreases the deformations in this area. Figure 14 illustrates both rib and channel types as they may occur in the center housing geometry.



normal channel



stiffened channel

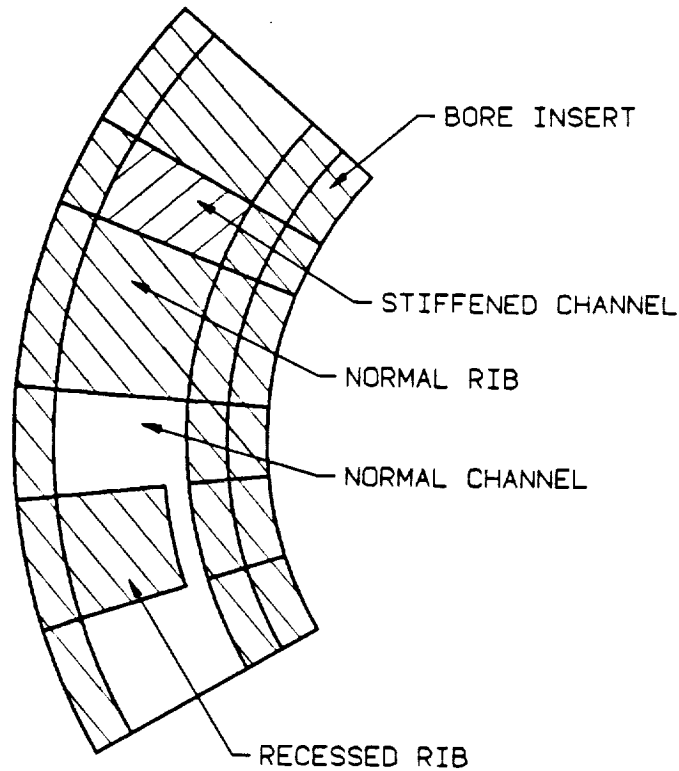
Figure 13. Modeling of the Two Channel Types

---

### 2.1.3 Bore Insert Definition

There has been increasing interest in converting the RCE to an AAD engine. To accommodate this interest, the SP will, if the user specifies, place a bore insert in the center housing. The bore insert could consist of a ceramic based material that reduces the heat rejection of the RCE.

In terms of the definitions of the building blocks of the housing geometry, the insert is the addition of a shell on the trochoidal surface. Instead of being defined as one inner shell, outer shell, and the material between shells, the rib is defined as two adjacent inner shells, an outer shell, and the material between the shells. The insert has a similar affect on the definition of a channel. (See Figure 14).



**Figure 14. Example of Combined Rib and Channel Types:** This figure illustrates how the rib and channel types can be combined to form the center housing geometry.

#### 2.1.4 Intake Port Definition

The intake port is a hole that runs radially through the center housing. At the inside edge of the center housing the hole is rectangular. At the outside edge, the hole is circular (Figure 15). The air and fuel mixture is injected from this port into the trochoidal chamber of the center housing. The SP has been written so that the intake port can be omitted from the center housing geometry. This option has been included

because in some RCE configurations, the intake port is located in the end plate, not the center housing.

### **2.1.5 Exhaust Port Definition**

The exhaust port, like the intake port, is a hole in the center housing that runs radially through the center housing. The hole is circular at the intersection of both the inside and outside edges of the center housing (Figure 16). The residue of the combustion process is ejected out of this port.

### **2.1.6 Spark Plug Port Definition**

The spark plug ports are, as expected, where the spark plugs are screwed into the center housing. Two spark plugs are used in the RCE to make the combustion process more efficient thus producing less exhaust emissions. The second spark plug burns some of the air fuel mixture not burned by the first spark plug. Like the intake and exhaust ports, the spark plug ports are holes that run radially through the center housing (Figure 17).

## **2.2 Geometric Parameter Definition**

The second step required in the development of the SP was to define the parameters that would be needed to fully describe the building blocks. In order that these user supplied parameters are as easily obtainable as possible, all the data required by the SP can

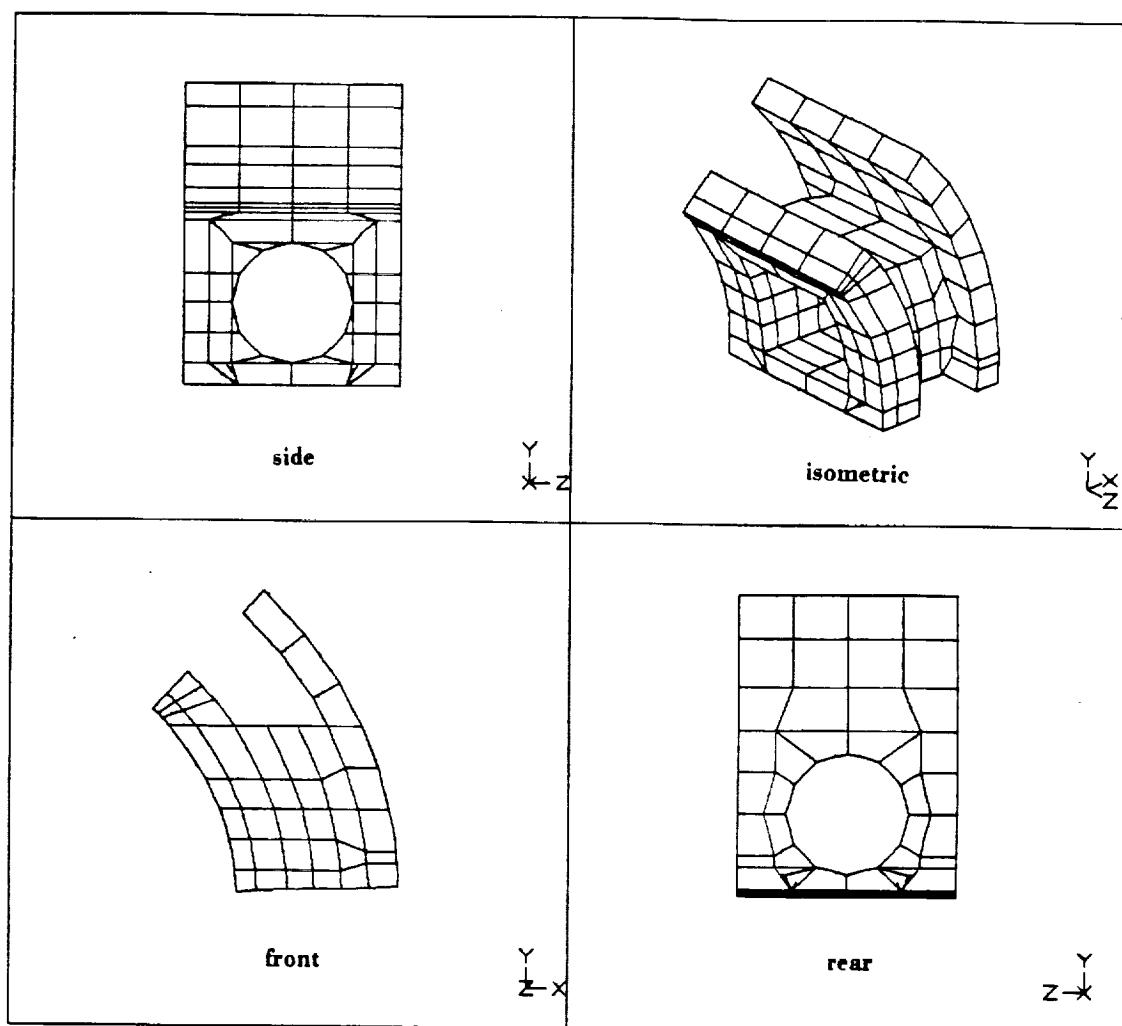
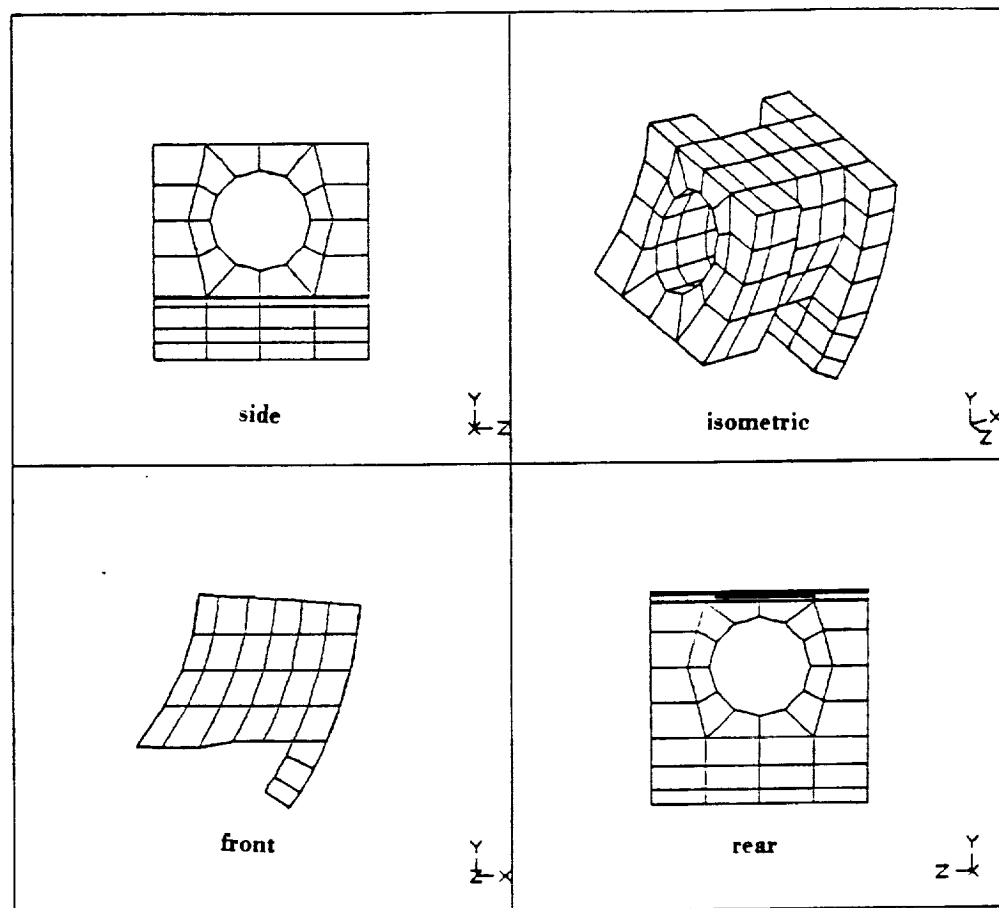
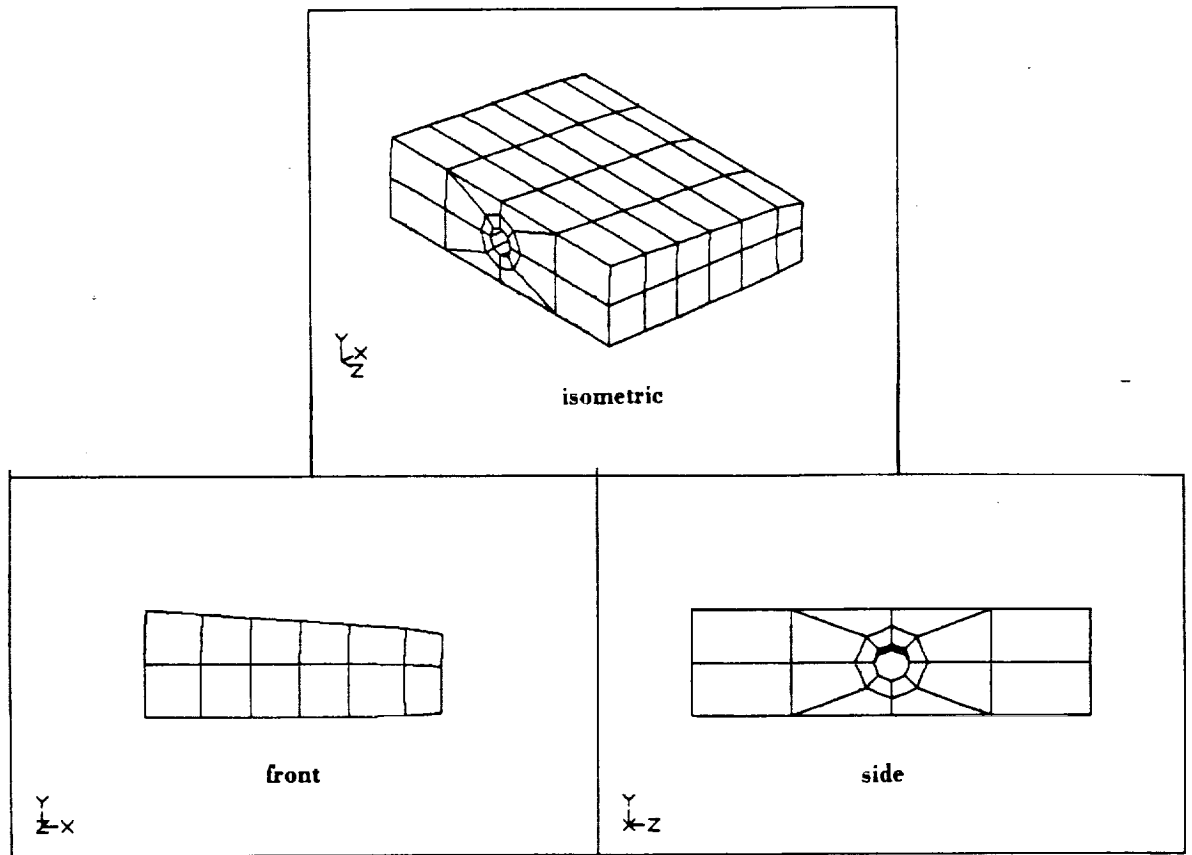


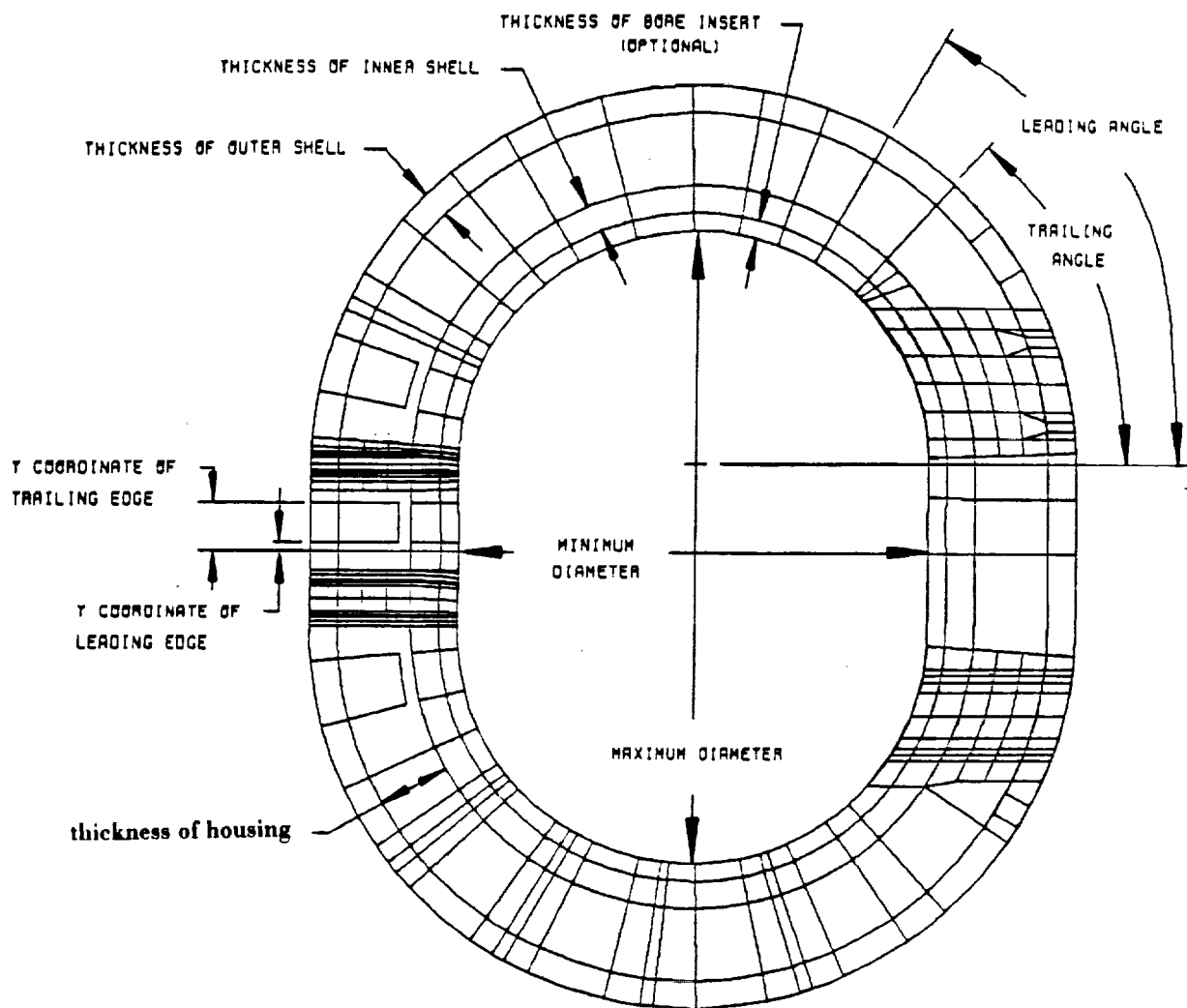
Figure 15. Modeling of the Intake Port



**Figure 16. Modeling of the Exhaust Port**



**Figure 17. Modeling of the Spark Plug Port**



**Figure 18. Typical Drawing of the Center Housing:** All the parameters needed to describe the housing geometry can be measured off of this drawing.



The inputs supplied by the user are:

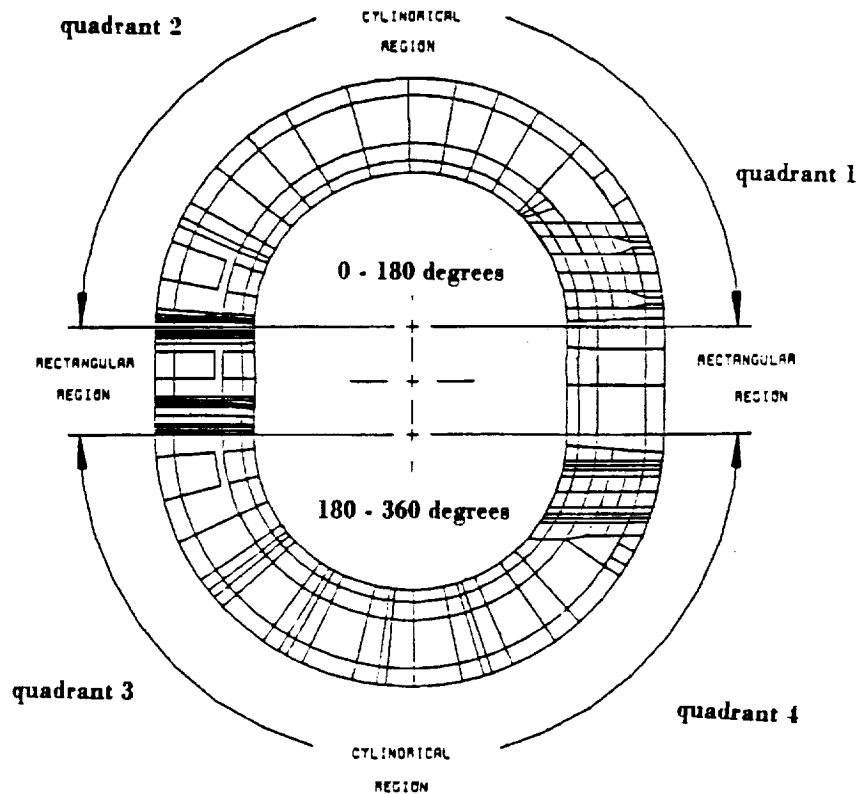
- The maximum and minimum diameters of the trochoidal bore.
- The thickness and axial depth of the center housing.
- The thickness of both the inner and outer shells.
- The thickness of the bore insert. (optional)
- The size, type and location of each rib.
- The type of each channel.
- The size and location the intake port.
- The size and location of the exhaust port.
- The size and location of the spark plug ports.

be measured from a drawing of the center housing. All parameter definitions are illustrated in Figure 18.

A detailed explanation of these parameters can be found in following sections.

The manner in which the user supplied parameters are defined changes depending on which region of the center housing is being constructed. There are two types of regions in the center housing - cylindrical and rectangular (Figure 19). In the cylindrical region, the inputs are in polar coordinates and in the rectangular region, the inputs are in Cartesian coordinates. Different coordinate systems are used so that the different parameters can be more easily obtained by the user.

There is a possibility that a rib or channel lies, not entirely within the cylindrical or rectangular region, but overlaps the two regions. When this overlapping of regions occurs, the SP must know what percent of the rib or channel lies in each region. The percentage of overlap must be known so that the SP can use the appropriate coordinate system - polar or Cartesian - to calculate the point coordinates of the overlapping rib or



**Figure 19. Illustration of the Different Regions in the Center Housing**

channel. The amount of overlap is calculated from the user supplied leading and trailing edge input that defines the location of the ribs and channels.

The overlap conditions were classified into six categories.

The overlap categories are:

1. Twenty percent or less of the **rib** lies in the cylindrical region
2. Eighty percent or more of the **rib** lies in the cylindrical region
3. Between twenty and eighty percent of the **rib** lies in the cylindrical region
4. Twenty percent or less of the **channel** lies in the cylindrical region
5. Eighty percent or more of the **channel** lies in the cylindrical region
6. Between twenty and eighty percent of the **channel** lies in the cylindrical region

Based upon the category that the overlapping rib or channel falls under, the SP calls a routine that uses the appropriate coordinate system to calculate the point coordinates that lie in the different regions.

### 2.2.1 Rib and Channel Input Parameter Definition

Because there are two different regions, the first input the user must specify is in which of these regions the rib lies or whether the rib overlaps the regions. If it is in the rectangular region, the y-coordinate of the leading and trailing edges must be specified. The x-coordinate is unnecessary because it is required that the ribs must be specified in a counterclockwise order starting in the first quadrant. This allows the x-coordinate to be defined by the housing input data. In the cylindrical region, the user must specify the angles of the leading and trailing edges of the rib. The last input needed to describe a rib is its type - normal or recessed. If a rib is recessed, the user must specify the magnitude of the recess.

Because the dimensions of the channels are defined by the borders of the ribs, the only input that is needed to describe the channel is its type - normal or stiffened. If a

channel is stiffened, a thin plate is placed at its midplane blocking the coolant flow. The thickness of the plate does have to be input because the plate is modeled with thin-shell elements. For thin-shell elements the thickness does not have to be defined by the geometry. The thickness is, however, defined in the finite element input data file and it is considered in the creation of the element stiffness matrix.

### **2.2.2 Intake and Exhaust Port Input Parameter Definition**

In order to maintain a uniformity in the generation of the housing geometry, the center housing is initially constructed only with ribs and channels. The intake and exhaust ports are then inserted into the channels specified by the user. When the ports are inserted, they do not occupy the entire channel; a small portion of the channel remains. To define where in the channel the ports are to be inserted and how much of the channel is to remain, the user must specify the leading and trailing angle of the ports at the intersection of the outer edge of the center housing (see Figure 20 for an illustration of these angles). The outer edge of the housing is chosen because the radius to this edge is known; the radius to the trochoidal edge is not. This trochoidal radius must be known when the input angles are being reduced into a form that can be used in the equations that calculate the point coordinates of the ports. These angles, as well as locating the port within the channel, also define the diameter of the circular end of the port. For the intake port, because one end of the port hole is rectangular, the user must specify the length and width of the rectangular hole. The last parameter needed to describe the intake or exhaust port is a wall thickness of the port. This input is needed because, in the axial direction, the ports are not flush with the edges of the housing. The indentation created by the port not being flush with the housing is termed the wall thickness of the port.

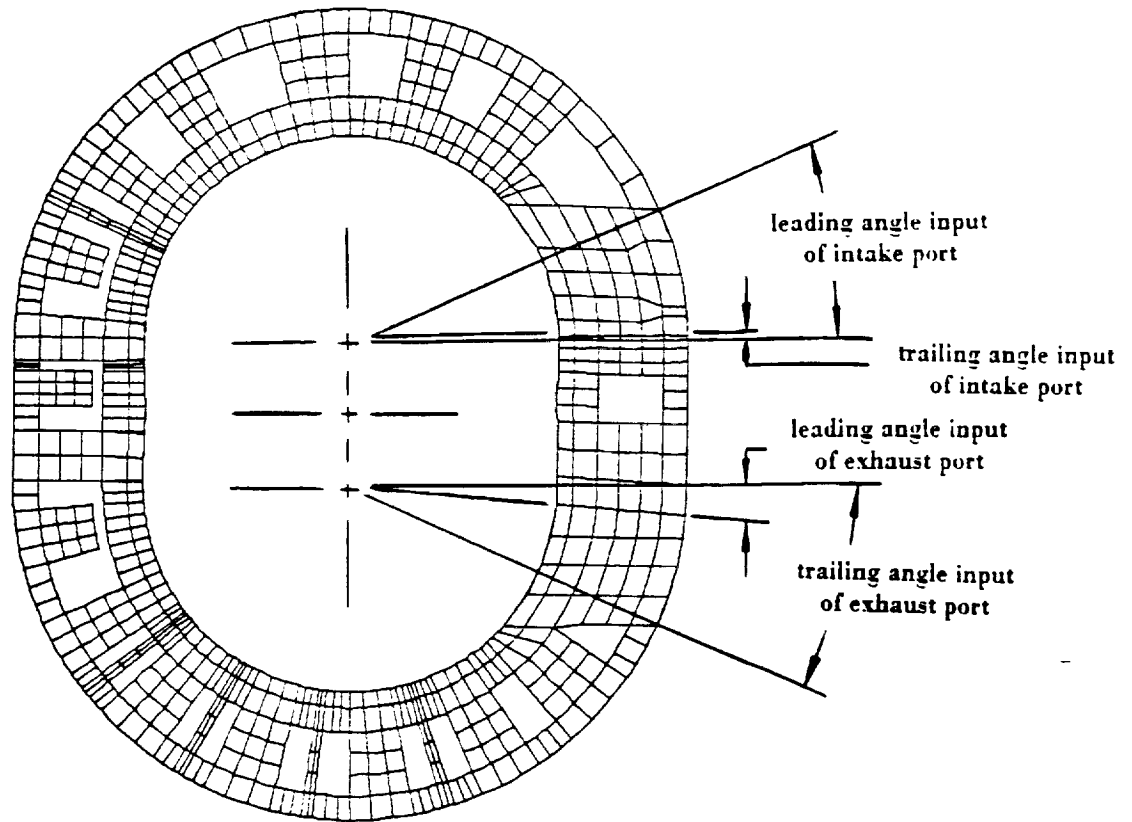


Figure 20. Illustration of the Intake and Exhaust Port Input Data

### 2.2.3 Spark Plug Port Input Parameter Definition

The spark plug can be thought of as a specialized rib: it is a rib that has a hole through it. The user has to specify which rib location is to have a circular hole in it and the radius of that hole. The user does not have to specify the location of the hole in the rib; it is always centered. The program has been written so that any number of spark plugs can be specified. Some RCE configurations have one spark plug and others have

two.

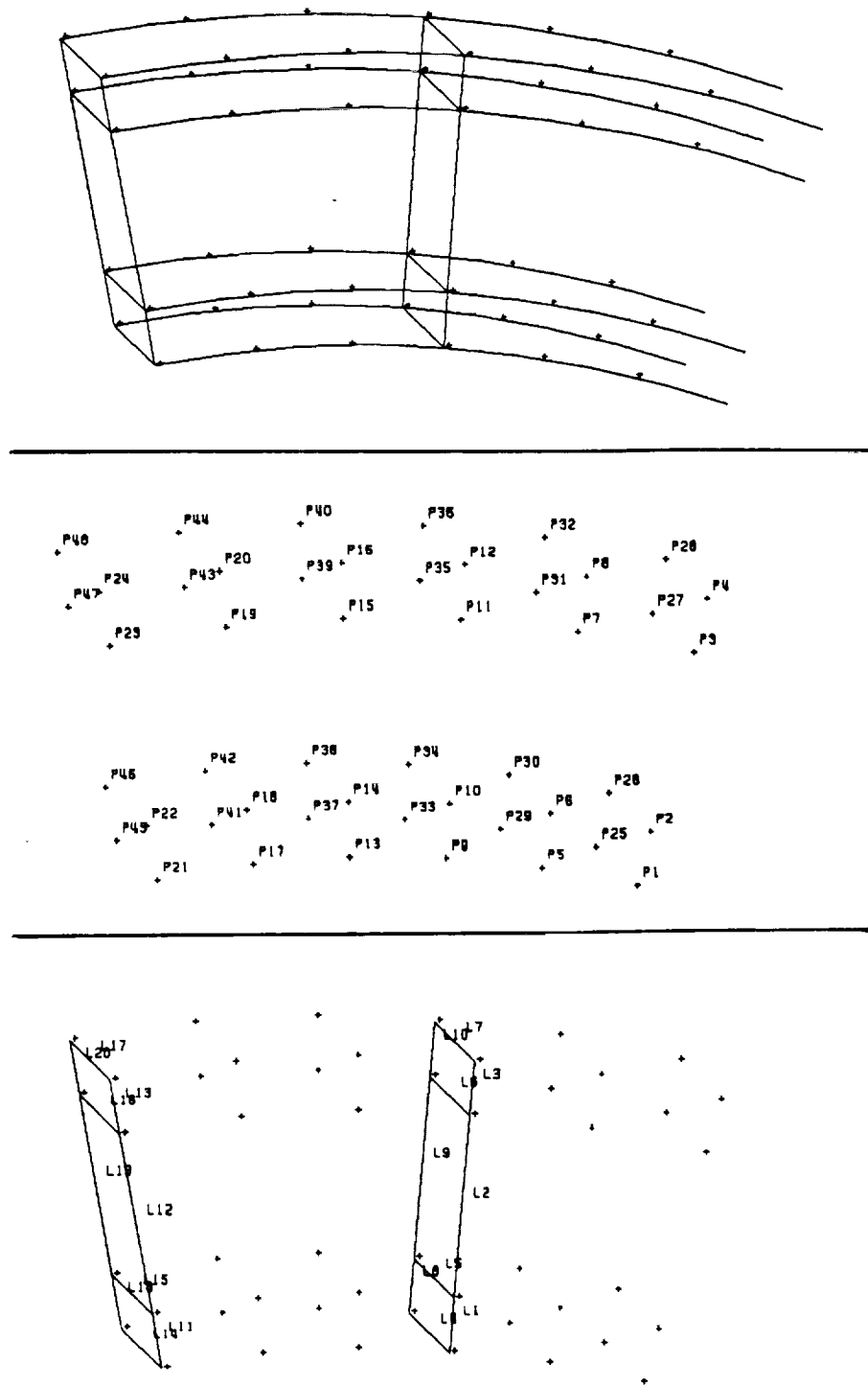
### 2.3 Numbering Patterns of the Geometric Entities

The difficulty in developing the UF arises in keeping the program general. All possible geometric configurations of the housing must be considered. The different types of ribs and channels must be able to be input in any combination. For example, one section of the housing might consist of a normal rib joined to a stiffened channel joined to a second normal rib while another section of the housing could consist of a different pattern of alternating rib and channel types. All possible combinations of the building blocks must be accounted for in order that their individual geometric entity label numbering patterns can be written into the UF. The numbering patterns of the geometric entity labels result from the order in which the geometric entities are used to create the building blocks. The generation order of the geometric entities that was chosen to create the building blocks progressed first radially, then axially, and finally angularly. This generation order created more repetitive numbering patterns than any other generation order. The more repetitive the numbering patterns were, the easier it was to organize them into a FORTRAN program (Figure 21 and Figure 22). It is critical that the numbering patterns of the entities be regular so that the UF maintains its ability to generate any user specified housing geometry.

To minimize the number of different numbering patterns needed to define the housing geometry, the ribs and channels were constructed together as rib/channel segments. There are three types of rib/channel segments - normal rib/normal channel, normal rib/stiffened channel, and recessed rib/normal channel (Figure 23). Associated with each of these three segments is a numbering pattern of the geometric entity labels. The user specifies which type of segment they are building and the SP calls the correct

line label	point labels	
	start	stop
1	9	10
2	10	11
3	11	12
4	9	33
5	10	34
6	11	35
7	12	36
8	33	34
9	34	35
10	35	36
11	21	22
12	22	23
13	23	24
14	21	45
15	22	46
16	23	47
17	24	48
18	45	46
19	46	47
20	47	48

**Figure 21. Example of a Numbering Pattern:** This pattern creates the lines of a normal rib/normal channel segment



**Figure 22. Pictures Generated from Numbering Pattern:** The top figure is a reference for the the bottom two figures. The middle figure is points and their labels. The bottom figure is the lines generated by the numbering pattern of the point labels.



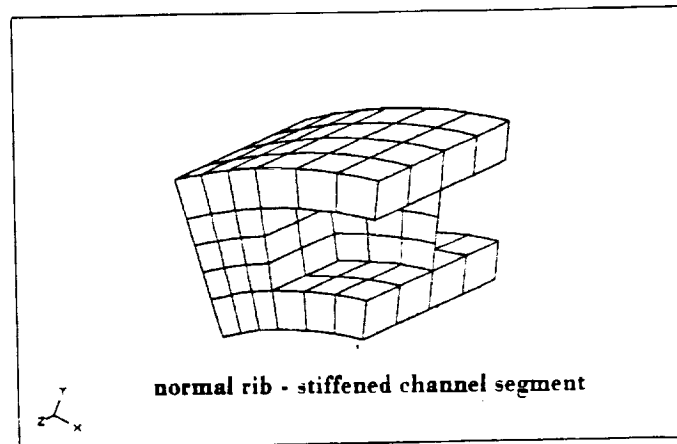
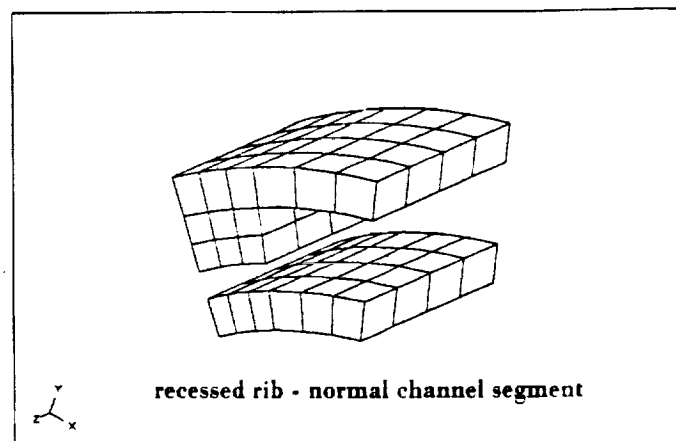
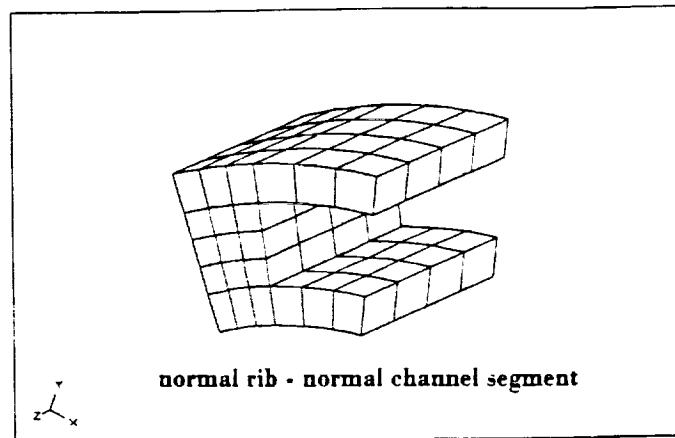


Figure 23. Illustration of the Three Rib/Channel Segment Types

## 2.4 Reduction & Implementation of Input for Universal File Generation

The input parameters can be categorized into three types. The first type are inputs that can be used directly by the SP, no conversion is necessary. For example, the axial depth of the housing is used unchanged to give the x and y point coordinates that are calculated by equations 1.1a and 1.1b a z dimension (depth). The second type of input is used to set flags in the SP. All the rib and channel type inputs fall into this category. For example, since different routines in the SP are used to generate the two rib types - normal and recessed - flags have to be used to ensure that the correct routine is called. The third and final type of input must be converted into a form that the SP can use to calculate the point coordinates of the geometry. For example, the maximum and minimum diameter inputs that define the trochoidal bore must be converted to the radius and eccentricity that are used in equations 1.1a and 1.1b. The following sections outline the necessary reductions for the third type of input.

### 2.4.1 Housing Dimension Input Reduction and Implementation

The housing parameters are used to calculate the constants, radius, eccentricity and perpendicular distance, that are used in equations 1.1a and 1.1b to define the trochoidal shape. The maximum and minimum diameters of the housing are used to calculate the radius and eccentricity of the trochoidal surface by applying the boundary conditions:

$$\text{minimum diameter} = \text{radius} - \text{eccentricity} \quad \text{when } \alpha = 90^\circ \quad (2.4.1b)$$

$$\text{maximum diameter} = \text{radius} + \text{eccentricity} \quad \text{when } \alpha = 0^\circ \quad (2.4.1a)$$

to equations 1.1a and 1.1b. Note that the perpendicular distance, 'A' in equations 1.1a and 1.1b, was set equal to zero for the generation of the inner edge of the trochoidal surface. This is why it does not appear in equations 2.4.1a and 2.4.1b. The thicknesses of the inner shell, outer shell and the housing that the user supplies define the different perpendicular distances that are used to generate the family of trochoidal curves that describe the housing geometry. The last housing parameter, the axial depth, is used to project the two dimensional model into three dimensions.

#### 2.4.2 Cylindrical Region Input Reduction and Implementation

To define a rib/channel segment that lies in the cylindrical region or overlaps the cylindrical and rectangular regions, the user supplies the angles of the leading and trailing edges of the rib. The leading and trailing angles, called phi, are measured from the center of the cylindrical region. The cylindrical region, as its name implies, is cylindrical. Because it is cylindrical, the center of the region can be located by subtracting the maximum y-coordinate of the trochoidal surface from the maximum x-coordinate. The difference of the two coordinates is the distance measured along the y-axis between the origin of the center housing and the center of the cylindrical region. (Figure 24).

- The maximum y-coordinate is easily determined; it is the radius plus the eccentricity of the trochoid. The x-coordinate was more difficult to determine. A numerical routine was written that, starting from zero and incrementing by one tenth of a degree to ninety degrees, would input alpha into equation 1.1a. The x-coordinates from equation 1.1b were compared until the largest value was found. Only one quadrant had to be

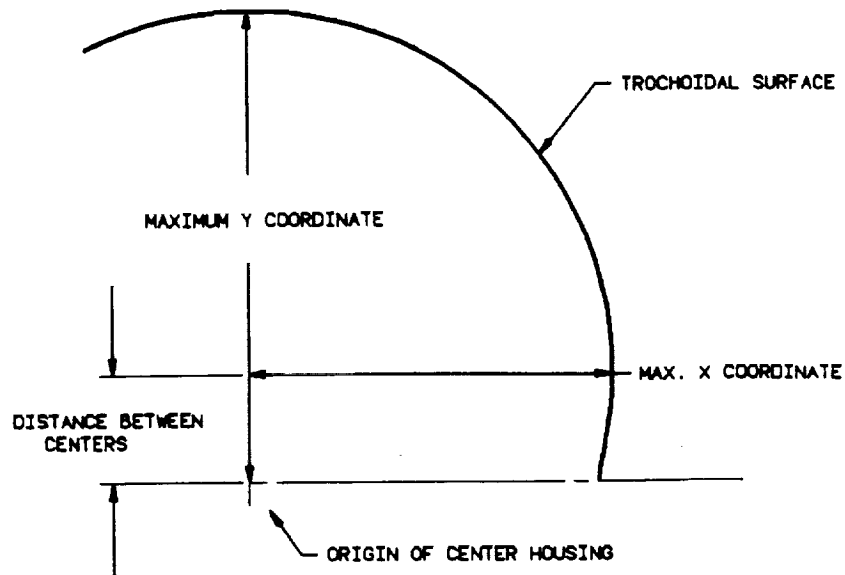


Figure 24. Determination of the Center of the Cylindrical Region

checked for the maximum x-coordinate since the trochoidal surface is symmetric about both axes.

The input angles,  $\phi$ , must be converted to the angles,  $\alpha$ , that are used in equations 1.1a and 1.1b. To accomplish this conversion, a numerical procedure was written that calculates an  $\alpha$ , given a  $\phi$ . The procedure uses the incremental search method to 'trap' the solution between an upper and lower value and the bisection method to solve for the root explicitly. The bisection method solves the equation:

$$\alpha = \tan(\phi) \times [(E \times \cos(3\alpha) + R \times \cos(\alpha)) - (E \times \sin(3\alpha) + R \times \sin(\alpha))] + T \quad (2.4.2a)$$

where E and R are the eccentricity and radius of the housing respectively, and T is the distance between centers. The bisection method was used because its convergence to the root of the equation was not affected by the inflection points that occur in the plot of alpha versus phi (Figure 25). So, given an angle phi, a corresponding angle alpha can be solved for. Alpha is used in equations 1.1a and 1.1b to solve for the x and y coordinate of a point on the trochoidal surface.

Now, after alpha has been obtained, theta is the only unknown in equations 1.1a and 1.1b. Theta is given by:

$$\theta = \cos^{-1} \left[ \frac{(R + 3 \times \cos(2 \times \alpha))}{\sqrt{9 \times E^2 + R^2 + 6 \times E \times R \times \cos(2 \times \alpha)}} \right] \quad (2.4.2b)$$

After theta is known, the x and y coordinates of the trochoidal surface can be calculated. The x and y-coordinates are given a z-coordinate equal to the axial depth to create three-dimensional point coordinates.

Seven different values of phi are needed to calculate the point coordinates that describe a rib/channel segment. Two of the seven values of phi come from the leading and trailing angle input. A third comes from the leading angle of the previous rib. The remaining four values of phi are calculated by dividing both the rib and channel into three equal parts (Figure 26). The seven values of phi are used to calculate seven two-dimensional point coordinates on the outside of the housing. The values of phi are converted to values of alpha by the numerical routine previously mentioned. The values of alpha are used to solve for values of theta and then both the values of alpha and theta are used to calculate seven two-dimensional point coordinates on the trochoidal surface. To create three dimensional model, these fourteen two-dimensional point coordinates (seven describing the outside edge of the housing and seven describing the trochoidal surface) are given a third dimension equal to the axial depth of the center housing. This brings the

## ILLUSTRATION OF THE INFLECTION POINTS

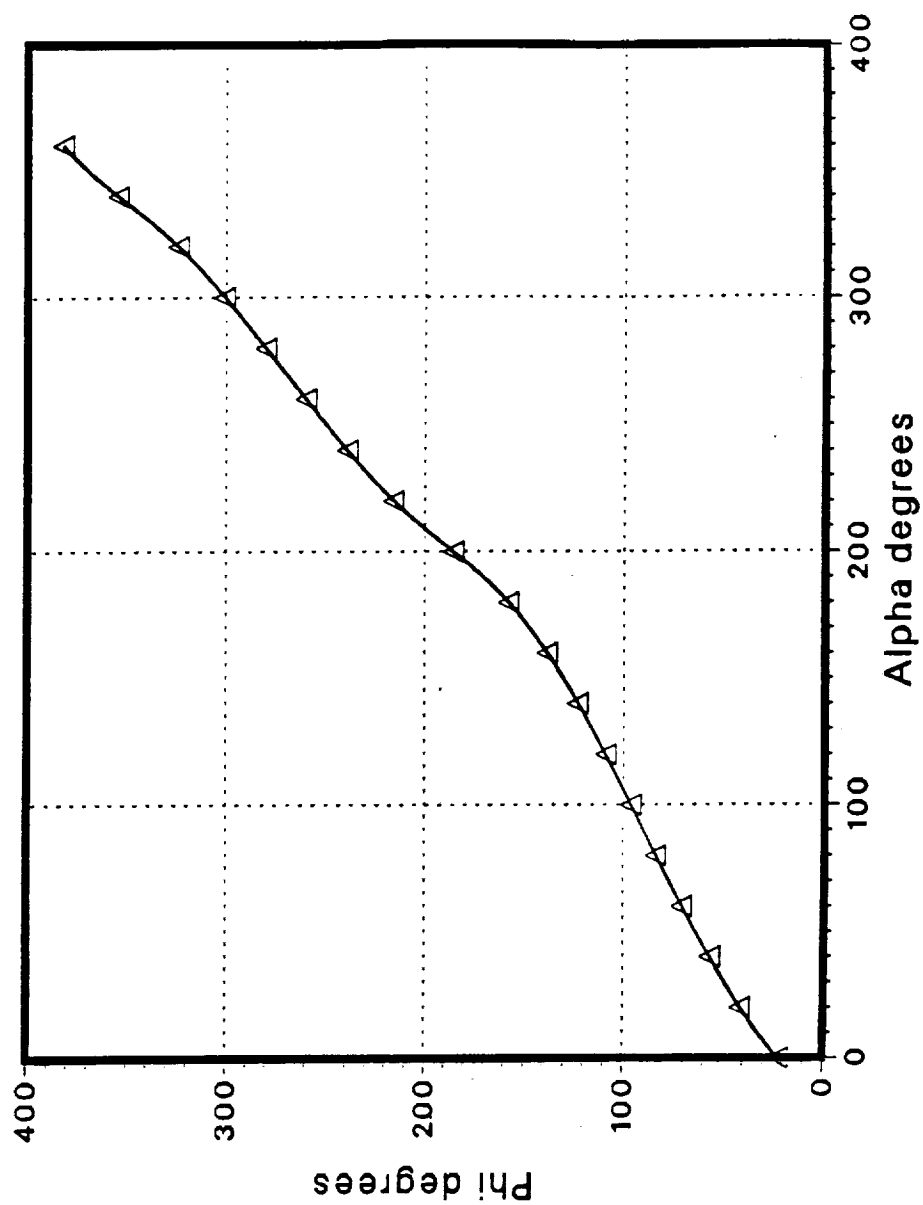
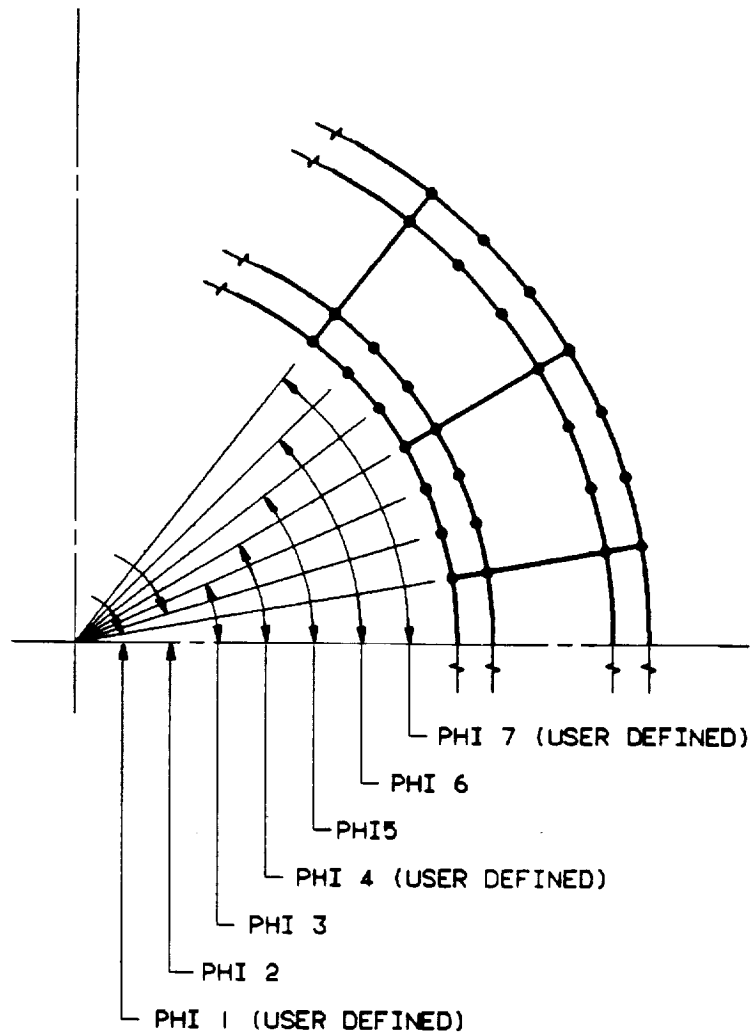


Figure 25. Graph of Alpha versus Phi



NOTE:  $\text{PHI } 2 = \text{PHI } 1 + (\text{PHI } 4 - \text{PHI } 1)/3$   
 $\text{PHI } 3 = \text{PHI } 1 + 2(\text{PHI } 4 - \text{PHI } 1)/3.0$   
 $\text{PHI } 5 = \text{PHI } 4 + (\text{PHI } 7 - \text{PHI } 4)/3$   
 $\text{PHI } 6 = \text{PHI } 4 + 2(\text{PHI } 7 - \text{PHI } 4)/3$

**Figure 26. Illustration of the Seven Phi Angles:** In the cylindrical region, seven phi angles are needed to define a rib/channel segment.

total number of points that describe a normal rib - normal channel segment to twenty eight.

Thirty-six point coordinates are needed to describe a rib/channel segment if either the rib is recessed or the channel is stiffened. To create the recessed rib, the eight point coordinates defining the trochoidal surface of the rib are radially expanded a distance equal to the magnitude of the recess. This provides the gap between the rib and inner shell that allows for more coolant flow. To create the stiffened channel, the four two-dimensional point coordinates defining the trochoidal surface of the channel are given a third dimension equal to one half of the axial depth. This defines the bottom spline of the stiffening plate in the channel. To define the top spline of the plate, the four point coordinates on the outer surface of the center housing are also given a third dimension equal to one half of the axial depth.

### 2.4.3 Rectangular Region Input Reduction and Implementation

If the rib/channel segment lies entirely within the rectangular region, the user supplies the y-coordinate of the leading and trailing edges of a rib. To calculate the coordinates of the points on the trochoidal surface, alpha, of equations 1.1a and 1.1b, must be known. To solve for alpha, given a y-coordinate, a numerical procedure, was written. The procedure uses the incremental search and the bisection method to solve the equation:

$$\alpha = E \times \sin(3\alpha) + R \times \sin(\alpha) - YCOR \quad (2.4.3a)$$

where E and R are the eccentricity and radius of the center housing respectively and YCOR is the user supplied y-coordinate.

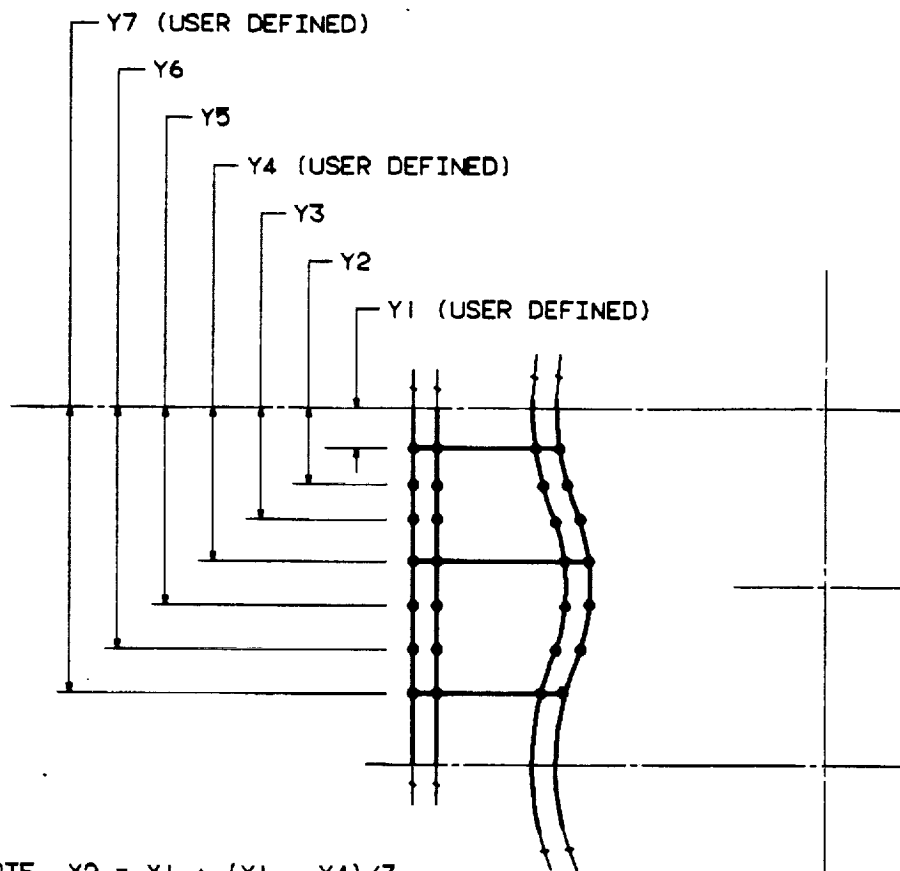


Seven y-coordinates are needed to define a rib/channel segment in the rectangular region. The seven y-coordinates are obtained in the same manner that was described for obtaining the seven phi angles. Two of the seven y-coordinates come from the leading and trailing edge input. A third comes from the leading edge of the previous rib. The remaining four y-coordinates are calculated by dividing both the rib and channel into three equal parts (Figure 27).

It takes 28 point coordinates to define a normal rib/normal channel segment. The labels of the 28 point coordinates are used in groups of two to define 30 lines. The point labels are also used in groups of four to define the 16 splines. Each line and spline label defines an edge. The edge labels are used in groups of four to define 29 surfaces. The surface labels are used in groups of six to define the five volumes of the rib/channel segment. It takes 8 more point coordinates, two more lines, two more splines, four more edges, and one more surface to define a normal rib/stiffened channel or recessed rib/normal channel segment than it does to define a normal rib/normal channel segment.

#### **2.4.4 Intake Port Input Reduction and Implementation**

To define the intake port, the user specifies into which channel the port is to be placed, the length and width of the rectangular hole that intersects the trochoidal surface, and the diameter of the circular hole that intersects the outer surface. The diameter of the circular hole is defined by the leading and trailing angle of the edges of the hole (See Figure 20). These inputs are used unaltered to calculate the point coordinates that are used to define the intake port. The first points calculated define the outer edge of the center housing. Because the port is oriented horizontally in the housing, the remaining points defining the port are calculated by subtracting from the X-coordinates defining the outer edge; the Y and Z coordinates remain constant. The distance subtracted from the



NOTE:  $Y2 = Y1 + (Y1 - Y4)/3$

$Y3 = Y1 + 2(Y1 - Y4)/3$

$Y5 = Y4 + (Y4 - Y7)/3$

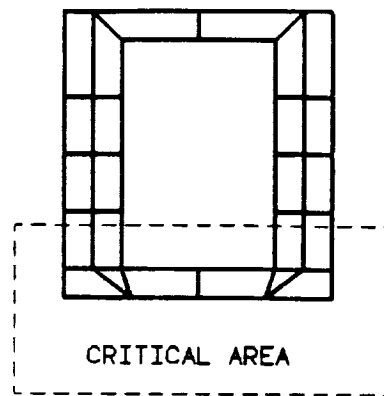
$Y6 = Y4 + 2(Y4 - Y7)/3$

**Figure 27. Illustration of the Seven Y-Coordinates:** In the rectangular region, seven y-coordinates are needed to define a rib channel segment.

X-coordinates is calculated from the user-supplied inputs that define the thicknesses of the housing and shells.

It takes 317 points to define the intake port. The numeric labels of the 317 points are used in groups of two to define 768 lines in the intake port. Each numeric label of a line defines an edge. The edge labels are used in groups of three or four to define the 596 surfaces of the intake port. The surfaces are used in groups of five or six to define the 150 volumes of the intake port.

Due to the complicated nature of the intake port geometry, irregularly shaped elements can be created for a given set of input dimensions. Distorted elements cause the finite element analysis to yield erroneous results. To eliminate the possibility of distorted elements being created, a critical area was isolated where highly distorted elements were likely to occur if certain input dimensions were used. The critical area was found at the bottom edge of the rectangular hole (Figure 28). To prevent the creation of distorted elements, the volumes were created at a size where only one element was placed on each volume. This allowed the elements in the critical areas to be created with triangular elements. The generated shape of the triangular elements was easier to control. Although distorted elements may still be created in this area, it is not as probable now that triangular elements are employed.



SIDE VIEW OF INTAKE PORT

Figure 28. Critical Area for Distorted Elements

---

#### 2.4.5 Exhaust Port Input Reduction and Implementation

To define the exhaust port, the user specifies into which channel the port is to be placed and the diameter of the port. The diameter of the port is defined by the leading and trailing angle of the hole (See Figure 20). These inputs are used unaltered to calculate the point coordinates that are used to define the exhaust port. The points are calculated first at the outer edge of the center housing and, working horizontally inward in a manner similar to that described for the intake port point calculation, to the trochoidal edge.

It takes 248 points coordinates to define the exhaust port. The numeric labels of the 248 points are used in groups of two to define 585 lines. Each numeric label of a line defines an edge. The edge labels are used in groups of four to define the 766 surfaces. The surfaces are used in groups of six to define the 108 volumes of the exhaust port.

#### **2.4.6 Spark Plug Port Input Reduction and Implementation**

To define the spark plug port, the user specifies in which rib the port is to be placed and the radius of the port. Even though the spark plug port can either lie within the rectangular region or overlap the rectangular and cylindrical regions, the user specifies the same inputs. These inputs are used unaltered to calculate the point coordinates of the port.

It takes 168 points coordinates to define each spark plug port. The numeric labels of the 168 points are used in groups of two to define 585 lines. Each numeric label of a line defines an edge. The edge labels are used in groups of four to define the 426 surfaces. The surfaces are used in groups of six to define the 96 volumes of a spark plug port.

## **CHAPTER 3: Program File Development**

The PF contains the commands that are used to guide the complicated algorithms in IDEAS through the generation of the meshes, nodes and elements. The problem encountered with creating the PF is the fact that the programmability allowed in IDEAS can not read externally defined variables. An externally defined variable is a variable that is defined in a program other than that which is currently being executed. This means that variables in IDEAS can only be assigned values by IDEAS. An example of externally defined variables that IDEAS must know before it can create the PF is the element type of a given mesh or the material type of a given element.

To overcome the limitations of the programmability in IDEAS, an innovative approach was used. It involved the using of the FORTRAN program to create the PF. The PF that the FORTRAN program file creates, looks exactly like a PF that would be created by using the programmability in IDEAS. The difference being, the FORTRAN program has read variables that were calculated in the UF to create the PF. The programmability in IDEAS does not have this capability.

### **3.1 Mesh Definition**

Before the meshes are defined, the PF must define the type of element to be placed on the mesh. Once the element type parameter has been defined, the algorithms in IDEAS are set to receive the first command that begins the mesh generation. The first command indicates which volume is having a mesh placed upon it. The remaining commands involve the number and placement of the elements that are to be generated on the mesh. (Note that the elements are not generated at this point, but the mesh generating

algorithms in IDEAS must know something about the elements before the meshes can be generated.) For each surface or volume that is to have a mesh placed on it, the PF contains a set of commands that indicates which surface or volume is currently having a mesh placed on it and the number and placement of the elements on the mesh. For a typical geometry, the PF creates approximately 600 meshes.

### 3.2 Node and Element Definition

Before the nodes and elements can be defined, certain parameters must be set. The parameters indicate that the nodes and elements are auto numbered by IDEAS and that nontriangular elements are to be generated. Also, the material constants - modulus of elasticity and poisson's ratio - and the physical properties of the elements must be provided. An example of a physical property is the thickness of a thin shell element. Once the parameters are set and the meshes are described, the nodes and elements can be defined. The first command in the PF that begins the node and element generation indicates which mesh is to have the nodes and elements placed upon it. The remaining commands in the PF set which material and physical properties are associated with the nodes and elements being generated. In summary, for each mesh, the PF contains a set of commands that indicates which mesh is currently having nodes and elements generated on it and which material and physical properties are associated with the nodes and elements. For a typical geometry, the PF creates over 3000 elements and 20,000 nodes.

## CHAPTER 4: User's Guide

The user supplied input needed by the SP were designed to be easily obtainable. To obtain the required input, the user needs either a drawing of the housing or an actual housing. It should be noted that the user does not have to obtain the input data before he/she executes the SP. The user can execute the SP and obtain each input as the SP prompts for it.

### 4.1 Obtaining the Input

The first input required by the SP are the maximum and minimum diameters of the trochoidal bore. Next the thicknesses of the housing's, inner shell and outer shell is required. All of the input can be measured directly off the housing or drawing such as Figure 18 on page 27. This input, along with any other input defining length, width or depth, must be input in inches.

Before the user can continue obtaining input data, the center of the cylindrical regions, shown in Figure 19 on page 29, must be located. The distance from the origin, the geometric center of the trochoid, to the center of the cylindrical regions is one half of the difference between the maximum and minimum diameter. For example, if the maximum diameter was 8.0 and the minimum diameter was 6.0, then the distance from the origin to the centers of the cylindrical regions would be:

$$1.0 = \frac{(8.0 - 6.0)}{2.0} \quad (4.1)$$



Now that the centers of the cylindrical region has been established, the data describing the ribs, channels and ports can be measured and input. The SP requires that the user begin the rib/channel input with the rib/channel segment which overlaps the intersection of the rectangular and cylindrical region in the first quadrant (see Figure 19 on page 29). The data for each remaining rib/channel segment must be input proceeding in a counterclockwise direction around the housing. This starting point and counterclockwise direction was chosen arbitrarily as no point or direction is better than any other.

The first input needed to describe a rib/channel segment is the region it lies in or whether it overlaps the two regions. Each region has been assigned a numeric label. The labels have been assigned as follows:

1. cylindrical region
2. rectangular region
3. overlaps the two regions

To specify a region, the user inputs the appropriate number associated with that region. Note that this input is in integer format (no decimal point).

The second input required to describe each rib/channel segment is the location of the leading and trailing edge of the rib. If the rib/channel segment overlaps the two regions or lies entirely within the cylindrical region, the leading and trailing edges are defined by the angle the edges make with respect to a horizontal line running through the center of the cylindrical region (see Figure 20 on page 32 - all angles must be measured in degrees). If the rib/channel segment lies entirely within the rectangular region, the rib is described by the y-coordinates of its edges. See Figure 18 on page 27 for an illustration of these inputs.

The third input required to describe a rib/channel segment is the type of the rib and channel. There are three rib types - normal, recessed, or a spark plug port - and there are

four channel types - normal, stiffened, intake port or exhaust port. Each rib and channel type has been assigned a numeric label. The labels have been assigned as follows:

1. normal rib
2. recessed rib
3. spark plug port rib

1. normal channel
2. recessed channel
3. exhaust port channel
4. intake port channel

To specify a certain rib or channel type, the user inputs the appropriate number associated with that type. Note that this input is in integer format (no decimal point).

If the rib or channel is not normal, then input other than the location of the leading and trailing edges is required. If the rib is recessed, then the magnitude of the recess must be measured and input. If the rib is a spark plug port, then the radius of the port must be input. If the channel is an exhaust port, then the radius of the port must be supplied. The radius of the ports is defined by angles similar to those shown in Figure 29. If the channel is an intake port, then the length and width of the rectangular portion of the port must also be specified.

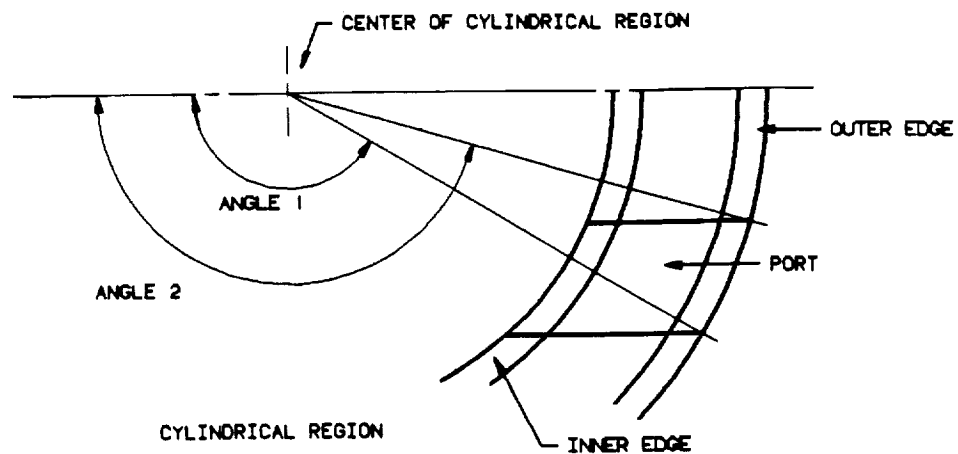


Figure 29. Angles Defining Radius of the Intake or Exhaust Port

## 4.2 Input Data File

The user has the choice of either answering the prompts as they appear on the screen or assembling a data file with all the input. If the user chooses to answer the prompts as they appear on the screen, then a data file of these responses will be created. This data file can be used on later executions of the SP. If the data file of input exists, then the prompts will not appear on the screen; the input needed for the SP will be read from the data file.

## CHAPTER 5: Conclusions

The SP will automatically generate a FEM from a minimal number of user supplied inputs that describe a particular RCE center housing geometry. The SP, significantly reducing the many hours it takes to create a FEM, makes the design process and analysis of alternate housing designs more cost effective. Figure 30 and Figure 31 illustrate two different FEM's created by the SP.

The SP consists of almost 10,000 lines of FORTRAN code. The two files - the UF and the PF - that the SP creates, contain a total of approximately 25,000 lines of data. The UF consists of approximately 20,000 lines of data that describes the center housing geometry. The PF consists of approximately 3000 lines of commands that guide the algorithms in IDEAS through the generation of the meshes nodes and elements.

It takes an IBM 4341 computer approximately seven hours to generate the FEM. This total execution time is significantly affected by the system utilization while the SP is being executed. A majority of the execution time - approximately eighty percent - is used to generate the nodes and elements. Approximately ten percent of the execution time is used to generate the meshes. The remaining approximate ten percent of the execution time is used to execute the SP, read the UF and PF, store files, and perform other miscellaneous tasks. The user does not have to be present during the execution time. He/she has only to input the parameters that define the housing geometry and the SP automatically generates the finite element model.

The SP is limited in one key area; the number and placement of the nodes and elements is not variable. Two steps can be taken to reduce this limitation. The step first involves using the option of exiting the SP at any point during its execution. This gives the user the freedom to exit the SP after the generation of the geometry so that the nodes

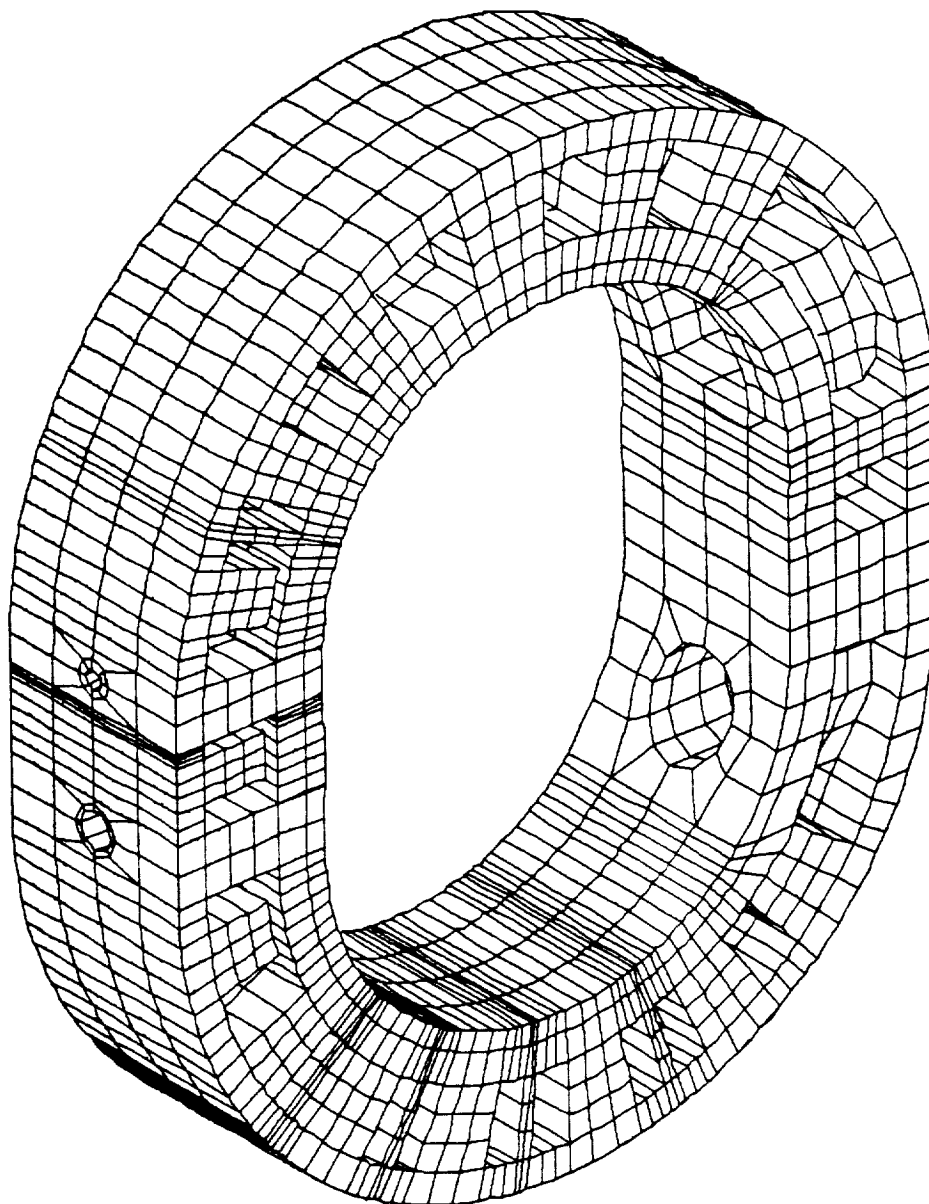
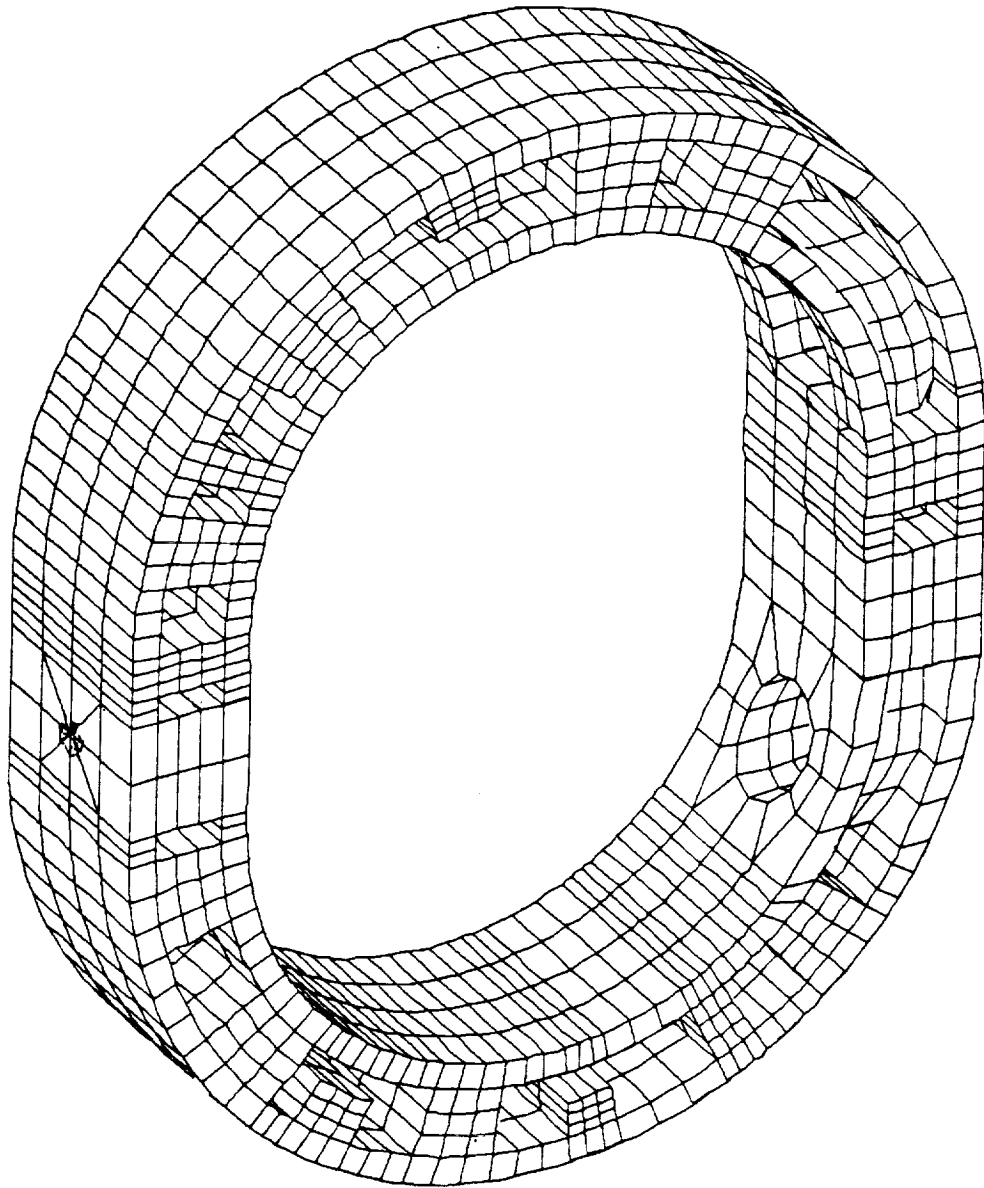


Figure 30. FEM Generated by the SP

---



**Figure 31.** A second FEM Generated by the SP: Note that there is no bore insert and the location, size and type of the rib/channel segments is different.

---

and elements can be generated manually. The opportunity to exit the SP at any point in the finite element model creation also allows the user to verify that the SP generated the desired geometry before the time consuming generation of the nodes and elements begins.

The second and more practical step that can be used by the user to reduce the limitation of the SP is to use the "add element" capability offered in the IDEAS software. The "add element" capability allows the user to create elements from existing nodes. The user can use the SP to create a FEM and then use the "add element" capability to refine the finite element mesh in areas of interest.

## References

1. Ansdale, A. F., *The Wankel RC Engine*, A. S. Barnes & Company, 1969.
2. Norbye, J. P., *The Wankel Engine: Design Development Applications*, Chilton Book Company, 1971.
3. Jones, C. & M. Berkwoitz, "Multifuel Rotary Aircraft Engine." SAE Paper No. 801237. Presented at AIAA SAE/ASME Joint Propulsion Congress, Hartford, CT, July, 1980.
4. Matsuura, K., K. Terasaki & I. Watanabe, "The Behavior of a Rotary Apex Seal Against the Trochoid Surface," Bulletin of the JSME, Vol. 21 No. 161, November 1978.
5. Eberle, M. K. & E. D. Klomp, "An Evaluation of the Potential Performance Gain from Leakage Reduction in Rotary Engines." SAE Paper No. 730117. Presented at the International Automotive Engineering Congress, Detroit, MI, January 10-15, 1973.
6. Yamamoto, K., K. Kenichi & K. Takashi, "Toyo Kogyo's Research and Development on Major Rotary Engine Problems," SAE Paper No. 700079. Presented at the International Congress & Exposition, Detroit, MI, January 12-16, 1970.
7. Burley, H. A., M. R. Meloeny & T. L. Stark, "Sources of Hydrocarbon Emissions in Rotary Engines," SAE Paper No. 780419. Presented at the International Congress & Exposition, Detroit, MI, February 27 - March 3, 1978.
8. Yamamoto, K. & T. Mukoki, "Development on Exhaust Emissions & Fuel Economy of the Rotary Engine at Toyo Kogyo," SAE Paper No. 780417. Presented at the International Congress and Exposition, Detroit, MI, February 27 - March 3, 1978.



9. Muroki, T., "Recent Technology Development of High-Powered Rotary Engine at Mazda," SAE Paper No. 841017. Presented at West Coast International Meeting & Exposition, San Diego, CA, August 6-9, 1984.
10. Jones, C. "An Update of Applicable Automotive Engine Rotary Stratified Charge Developments," SAE Paper No. 820347. Presented at the International Congress & Exposition, Detroit, MI, February 22-26, 1982.
11. Bradley, S. "Finite Element Model Development of a Rotary Combustion Engine Center Housing." A thesis presented at Michigan Technological University, Houghton, MI, February, 1985.
12. Krouse, J. K., "Reducing the Labor content in Finite Element Analysis," *Machine Design*, Vol. 19, No. 9, September 8, 1983, pp.64-69.
13. Bennett, J. A. & M. F. Nelson, "An Optimization Capability for Automotive Structures," SAE Paper No. 790972. 1979.
14. "User Manual for Supertab," Structural Dynamics Research Corporation, Vol 1, Level 2, 1983.
15. "Supertab User Manual Supplement," Structural Dynamics Research Corporation, Vol 1, Level 2, 1984.

## Appendix 1: Example of a Universally Formatted Data File

This universal file generates the cube pictured in Figure 32

-1		rectangular		color	coordinates
15	label	coordinate system			
	1	0	0	11	-1.00E+00 -1.00E+00 0.00E+00
	2	0	0	11	-9.99E-01 -9.99E-01 4.99E-01
	3	0	0	11	-9.99E-01 -9.99E-01 9.99E-01
	4	0	0	11	-9.99E-01 -9.99E-01 1.50E+00
	5	0	0	11	-9.99E-01 -9.99E-01 2.00E+00
	6	0	0	11	-9.99E-01 -9.99E-01 2.50E+00
nodes					
	688	0	0	11	9.99E-01 9.99E-01 7.50E-01
	689	0	0	11	9.99E-01 9.99E-01 1.00E+00
	690	0	0	11	9.99E-01 9.99E-01 1.50E+00
	691	0	0	11	9.99E-01 9.99E-01 2.00E+00
	692	0	0	11	9.99E-01 9.99E-01 2.50E+00
	693	0	0	11	9.99E-01 9.99E-01 3.00E+00
-1					

-1		rectangular		color	coordinates
25	label	coordinate system			
	1	0	7	-1.000E+00 -1.000E+00 0.000E+00	
	2	0	7	1.000E+00 -1.000E+00 0.000E+00	
	3	0	7	1.000E+00 1.000E+00 0.000E+00	
	4	0	7	-1.000E+00 1.000E+00 0.000E+00	
	5	0	7	-1.000E+00 -1.000E+00 3.000E+00	
	6	0	7	1.000E+00 -1.000E+00 3.000E+00	
	7	0	7	1.000E+00 1.000E+00 3.000E+00	
	8	0	7	-1.000E+00 1.000E+00 3.000E+00	
points					
-1					

-1		color	line style	point label	point label
26	label		(solid)	start	stop
	1	8	1	5	6
	2	8	1	6	7
	3	8	1	7	8
	4	8	1	8	5
	5	8	1	5	1
	6	8	1	6	2
	7	8	1	7	3
	8	8	1	8	4
lines					

	9	8	1	1	2
	10	8	1	2	3
	11	8	1	3	4
-1	12	8	1	4	1

			entity type
			1 line
		number of entities	2 are
	label	deining edge	3 spline
-1	29	color	
	1	9	1
	1	1	1
	2	9	1
	1	2	1
	3	9	1
	1	3	1
	4	9	1
	1	4	1
	5	9	1
	1	9	1
	6	9	1
edges	1	10	1
	7	9	1
	1	11	1
	8	9	1
	1	12	1
	9	9	1
	1	5	1
	10	9	1
	1	6	1
	11	9	1
	1	7	1
	12	9	1
-1	1	8	

			line style	number of edges	
			(dashed)	defining surface	
-1	30	color			
	1	4	2	4	
	4	1	2	3	
	2	4	2	4	
	9	1	10	5	edge labels
	3	4	2	4	
surfaces	10	2	11	6	
	4	4	12	4	
	11	7	2	3	
	5	4	2	4	
	12	8	9	4	
	6	4	2	4	
	5	6	7	8	
-1					

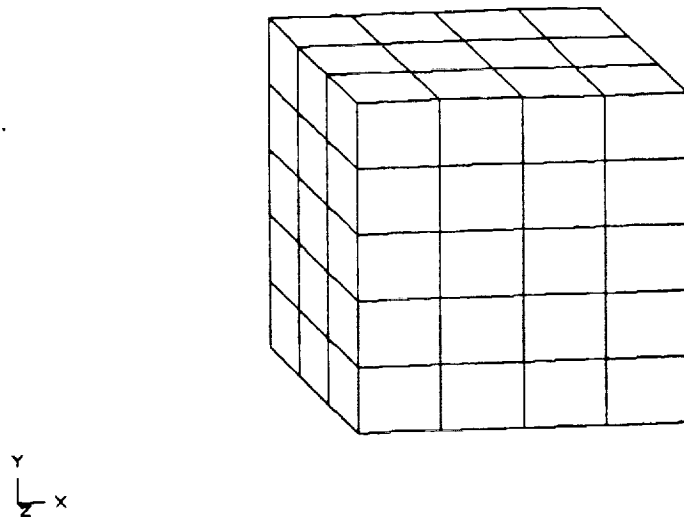
volumes	-1 39	label	color	line style (dashed)	number of surfaces defining volume	surface labels
		1 2	5 5	2 1	2 3	6 6
	-1					

label	element type parabolic solid	color	node labels
-1 71			
1	20	116	1
1	2	3	10
64	66	80	78
143	142	141	134
2	20	116	1
3	4	5	12
66	68	82	80
145	144	143	136
3	20	116	1
5	6	7	14
68	70	84	82
147	146	145	138

element connectivity

58	20	116	1	563	7	20	
547	548	549	556	563	562	561	554
610	612	626	624	673	674	675	682
689	688	687	680				
59	20	116	1	565	7	20	
549	550	551	558	565	564	563	556
612	614	628	626	675	676	677	684
691	690	689	682				
60	20	116	1	567	7	20	
551	552	553	560	567	566	565	558
614	616	630	628	677	678	679	686
693	692	691	684				
-1							

elements	label	element type parabolic solid	physical property	material property
-1 74				
	1	116	1	1
	2	116	1	1
	3	116	1	1
	58	116	58	1
	59	116	59	1
	60	116	60	1
-1				



---

**Figure 32. Example Model Generated by the Universal File**

---

Note that although there are nodes and elements defined in universal format, there are no meshes. The mesh data is used to create the nodes and elements. Once the nodes and elements are created, the mesh serves no purpose so it is not stored in universal format.

Note also, that the original universally formatted file contained none of the annotative comments. These comments have been added to give the reader a better understanding of the format.

## Appendix 2: Example of an Input Data File

This appendix is an example data file that creates a finite element model of the axially cooled rotary combustion engine center housing that was sent to Michigan Technological University by the NASA/ Lewis Research Center.

5.95,3.80	.....	maximum and minimum diameters
1.00	.....	thickness of housing
2.68	.....	depth of housing
0.375	.....	thickness of outer shell
0.375	.....	thickness of inner shell
0.25	.....	thickness of bore insert
3	.....	overlappin region
-5.0,2.0	.....	phi 1 and phi 2 of rib edges
1	.....	normal rib
2	.....	stiffened channel
0.75	.....	thickness of first channel
1	.....	cylindrical region
47.5,60.0	.....	phi 1 and phi 2 of rib edges
1	.....	normal rib
4	.....	intake port channel
2.0,23.0	.....	angles defining radius of port
0.275	.....	wall thickness of port
1.50,1.50	.....	length and width of port
1	.....	cylindrical region
70.0,80.0	.....	phi 1 and phi 2 of rib edges
1	.....	normal rib
2	.....	stiffened channel
1	.....	cylindrical region
91.0,105.0	....	phi 1 and phi 2 of rib edges
1	.....	normal rib
2	.....	stiffened channel
1	.....	cylindrical region
120.0,130.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib

2	.....	stiffened channel
1	.....	cylindrical region
140.0,150.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
2	.....	stiffened channel
1	.....	cylindrical region
154.0,156.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
2	.....	stiffened channel
1	.....	cylindrical region
160.0,169.0	...	phi 1 and phi 2 of rib edges
2	.....	recessed rib
0.175	.....	magnitude of recess
3	.....	overlapping region
176.0,184.0	...	phi 1 and phi 2 of rib edges
3	.....	rib is spark plug port
0.125	.....	radius of port
2	.....	rectangular region
0.60,0.05	.....	y coordinate 1 and y coordinate 2 of rib edges
2	.....	recessed rib
0.175	.....	magnitude of recess
2	.....	rectangular region
-0.33,-1.10	...	y coordinate 1 and y coordinate 2 of rib edges
3	.....	rib is spark plug port
0.25	.....	radius of port
3	.....	overlapping region
185.0,195.0	...	phi 1 and phi 2 of rib edges
2	.....	recessed rib
0.175	.....	magnitude of recess
1	.....	rectangular region
205.0,215.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	cylindrical region
218.0,220.0	....	phi 1 and phi2 of rib edges
1	.....	normal region
1	.....	normal channel
1	.....	cylindrical region
223.0,235.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
239.0,241.0	...	phi 1 and phi 2 of rib edges

1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
245.0,257.0	....	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
262.5,264.5	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
271.0,282.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
287.0,289.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
294.0,305.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
1	.....	cylindrical region
312.0,326.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
1	.....	normal channel
3	.....	overlapping region
356.0,365.0	...	phi 1 and phi 2 of rib edges
1	.....	normal rib
3	.....	channel is exhaust port
340.0,354.0	...	angles defining radius of port
0.275	.....	wall thickness of port
4	.....	end prompted input
1	...	entities to edges generated - continue model generation
1	...	entities to meshes generated - continue model generation



### **Appendix 3: Examples of Numbering Patterns**

The examples of the numbering patterns contained in this appendix are used in the SP to generate the geometric entities - splines, edges, surfaces, and and volumes - that are used to define a normal rib / normal channel segment.

### APPENDIX 3.1: Numbering Pattern of the Splines

spline label	point labels			
	pnt 1	pnt 2	pnt 3	pnt 4
1	x	1	5	9
2	x	2	6	10
3	x	3	7	11
4	x	4	8	12
5	x	25	29	33
6	x	26	30	34
7	x	27	31	35
8	x	28	32	36
9	9	13	17	21
10	10	14	18	22
11	11	15	19	23
12	12	16	20	24
13	33	37	41	45
14	34	38	42	46
15	35	39	43	47
16	36	40	44	48

**Figure 33. Numbering Pattern of the Splines:** this pattern creates a normal rib / normal channel segment

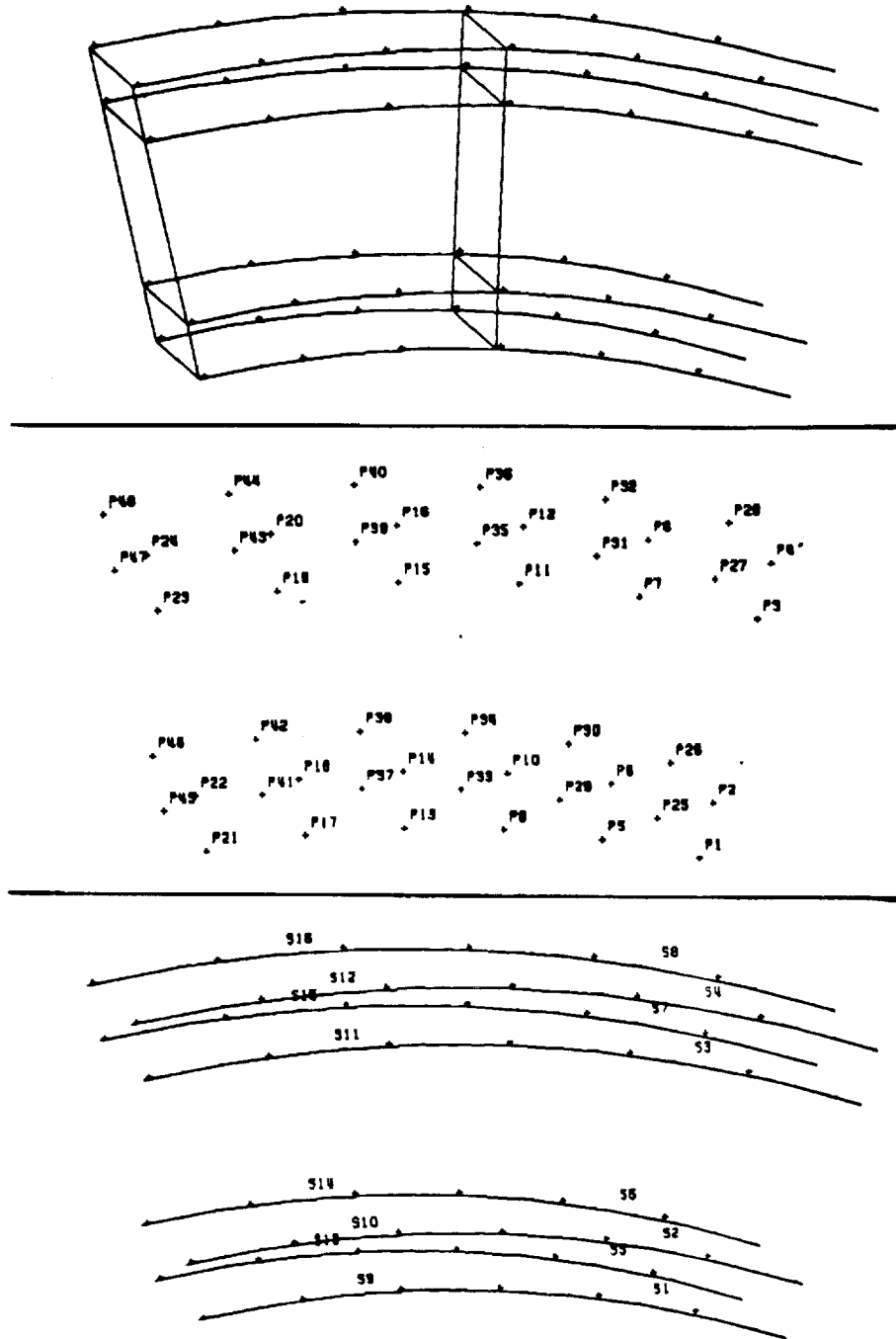


Figure 34. Illustration of the Numbering Pattern of the Splines

### APPENDIX 3.2: Numbering Pattern of the Edges

edge label	line & spline labels
1	spline 1
2	spline 2
3	spline 3
4	spline 4
5	spline 5
6	spline 6
7	spline 7
8	spline 8
9	line 1
10	line 2
11	line 3
12	line 4
13	line 5
14	line 6
15	line 7
16	line 8

17	spline 9
18	spline 10
19	spline 11
20	spline 12
21	spline 13
22	spline 14
23	spline 15
24	spline 16
25	line 9
26	line 10
27	line 11
28	line 12
29	line 13
30	line 14
31	line 15
32	line 16

**Figure 35. Numbering Pattern of the Edges:** this pattern creates a normal rib : normal channel segment

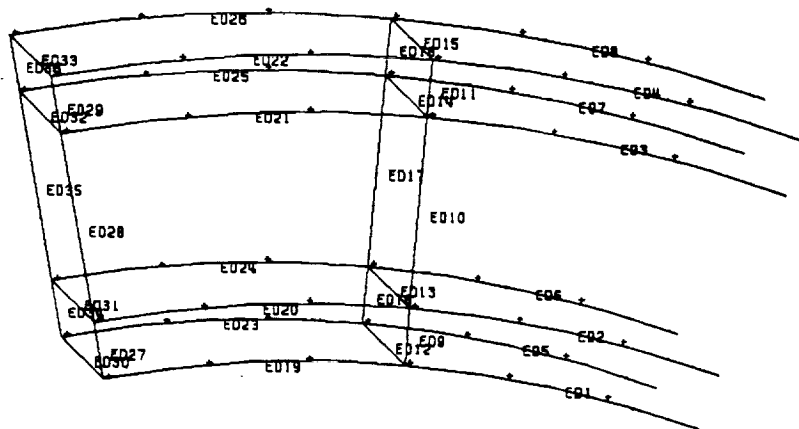
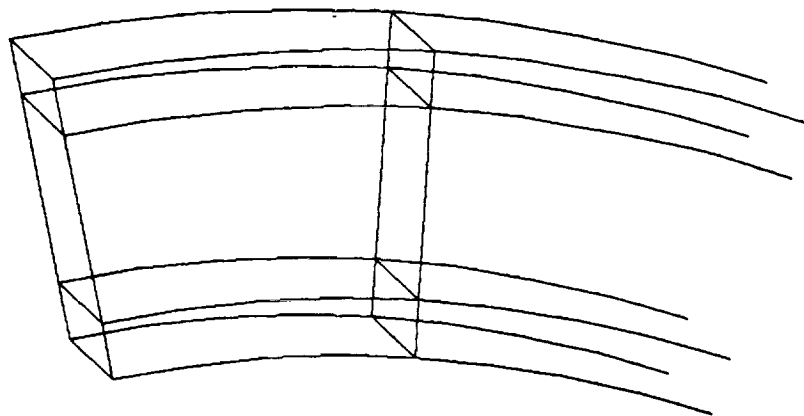


Figure 36. Illustration of the Numbering Pattern of the Edges

### APPENDIX 3.3: Numbering Pattern of the Surfaces

surface label	edge labels			
	edg 1	edg 2	edg 3	edg 4
1	1	x	2	9
2	3	x	4	11
3	1	x	5	12
4	2	x	6	13
5	3	x	7	14
6	4	x	8	15
7	5	x	6	16
8	7	x	8	18
9	12	9	13	16
10	13	10	14	17
11	14	11	15	18
12	19	9	20	27
13	20	10	21	28
14	21	11	22	29
15	19	12	23	30
16	20	13	24	31
17	21	14	25	32
18	22	15	26	33
19	23	16	24	34
20	24	17	25	35
21	25	18	26	36
22	30	27	31	34
23	31	28	32	35
24	32	29	33	36

**Figure 37. Numbering Pattern of the Surfaces:** this pattern creates a normal rib : normal channel segment

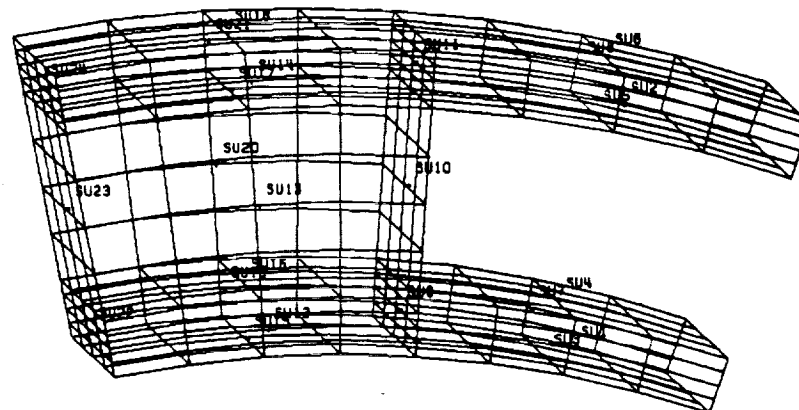
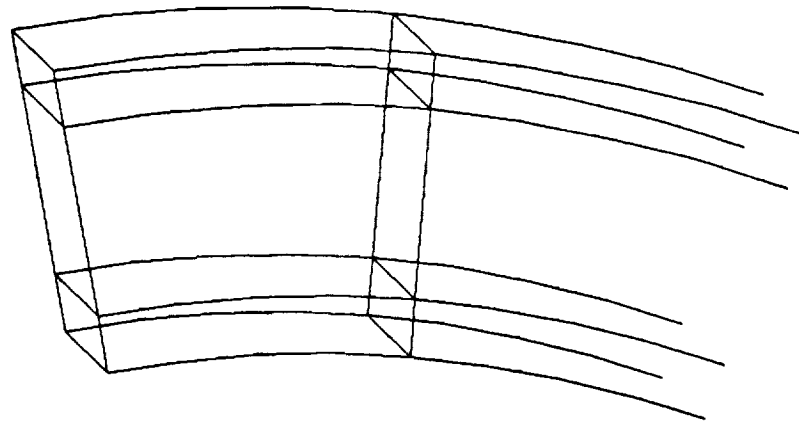


Figure 38. Illustration of the Numbering Pattern of the Surfaces

### APPENDIX 3.4: Numbering Pattern of the Volumes

---

volume label	surface labels					
	sur 1	sur 2	sur 3	sur 4	sur 5	sur 6
1	3	1	x	7	9	4
2	5	2	x	8	11	6
3	15	12	9	19	22	16
4	16	13	10	20	23	17
5	17	14	11	21	24	18

**Figure 39. Numbering Pattern of the Volumes:** this pattern creates a normal rib / normal channel segment

---



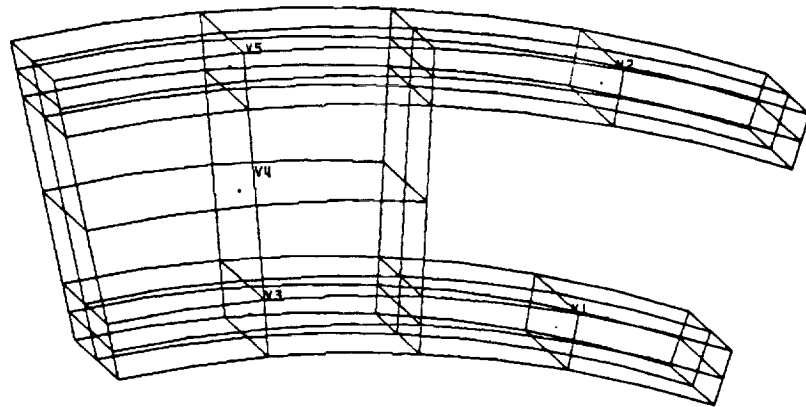
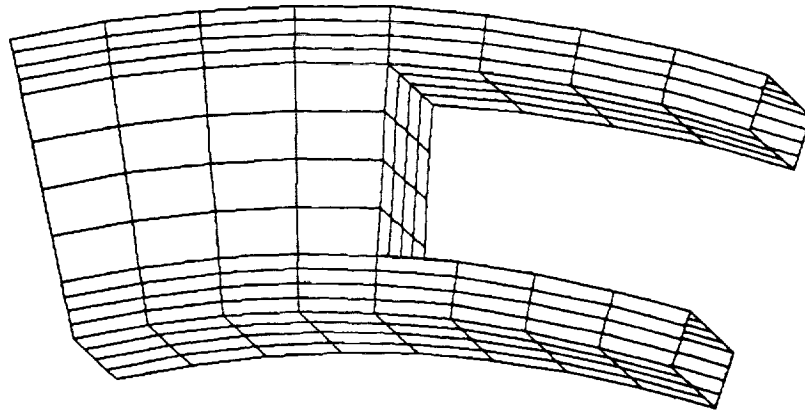


Figure 40. Illustration of the Numbering Pattern of the Volumes

# Appendix

C

```
C *****
C *
C *   THIS PROGRAM GENERATES FINITE ELEMENT MODELS OF
C *   USER DEFINED ROTARY COMBUSTION ENGINE CENTER
C *   HOUSING GEOMETRIES. THE USER IS PROMPTED FOR THE
C *   INPUT DATA. THIS PROGRAM WILL OUTPUT SEVERAL FILES.
C *   THE FILE CONTAINING THE DATA THE DESCRIBES THE GEOM-
C *   TRY IS CALLED "GEOMETRY". THE FILES CONTAINING THE
C *   COMMANDS THAT GENERATE THE MESHES, NODES AND ELEMENTS
C *   IS CALLED "MSNDEL".
C *
C *****
C
C *****
C *
C *   VARIABLE DEFINITION
C *
C *   THE VARIABLES THAT BEGIN WITH THE SAME FIRST THREE
C *   LETTERS ARE THE POINT COORDINATES OF THE BORE.
C *   INSERT (AAA - GGG). THE VARIABLES THAT BEGIN WITH
C *   THE SAME FIRST TWO LETTERS ARE THE POINT COORDINATES
C *   OF THE INNER EDGE OF THE INNER SHELL (AA - GG). THE
C *   VARIABLES THE BEGIN WITH THE LETTERS (A -G) ARE THE
C *   POINT COORDINATES OF THE OUTER EDGE OF THE INNER
C *   SHELL. THE VARIABLES THAT BEGIN WITH THE LETTERS
C *   (H - N) ARE THE POINT COORDINATES OF THE INNER EDGE
C *   OF THE OUTER SHELL. THE VARIABLES THAT BEGIN WITH
C *   THE LETTERS (HH-NN) ARE THE POINT COORDIANTES OF THE
C *   OUTER EDGE OF THE OUTER SHELL.
C *
C *****
C
C   DIMENSION AIX(100),AIY(100),BIX(100),BIY(100),CIX(100),CIY(100),
C # DIX(100),DIY(100),EIX(100),EIY(100),FIX(100),FIY(100),GIX(100),
C # GIY(100),HIX(100),HIY(100),IIX(100),IIY(100),JIX(100),JIY(100),
C # KIX(100),KIY(100),LIX(100),LIY(100),MIX(100),MIY(100),NIX(100),
C # NIY(100),AAIX(100),AAIY(100),BBIX(100),BBIY(100),
C # CCIX(100),CCIY(100),DDIX(100),DDIY(100),EEIX(100),EEIY(100),
C # FFIX(100),FFIY(100),GGIX(100),GGIY(100),HHIX(100),HHIY(100),
C # IIX(100),IIY(100),JJIX(100),JJIY(100),KKIX(100),KKIY(100),
C # LLIX(100),LLIY(100),MMIX(100),MMIY(1000),NNIX(1000),NNIY(100),
C # DRIX(100),DRIY(100),ERIX(100),ERIY(100),FRIX(100),FRIY(100),
C # GRIX(100),GRIY(100),REGION(100),RSP(10),PHI1SP(10),PHI2SP(10),
C # AAAIX(100),AAAIY(100),
C # BBBIX(100),BBBIY(100),CCCIX(100),CCCIY(100),DDDIX(100),DDDIY(100),
C # EEEIX(100),EEEIY(100),FFFFIX(100),FFFIY(100),GGGIX(100),GGGIY(100),
C # Y1SP(10),Y2SP(10),ISPRK(10),PHI1(1000),XX(1000),ZZZ(100),RZZZ(100)
C *****
C *
C *   THIS COMMON BLOCK CONTAINS THE POINT COORDINATES
C *   OF THE INTAKE, EXHAUST, AND SPARK PLUG PORTS.
C *
C *****
C
C   COMMON / PORT / X1(7),Y1(7),Z1(7),X2(7),Y2(7),Z2(7),X3(7),Y3(7),
C # Z3(7),X4(7),Y4(7),Z4(7),X5(7),Y5(7),Z5(7),X6(7),Y6(7),Z6(7),
C # X7(7),Y7(7),Z7(7),X8(7),Y8(7),Z8(7),X9(7),Y9(7),Z9(7),
C # X10(7),Y10(7),Z10(7),X11(7),Y11(7),Z11(7),X12(7),Y12(7),Z12(7),
C # X13(7),Y13(7),Z13(7),X14(7),Y14(7),Z14(7),X15(7),Y15(7),Z15(7),
C # X16(7),Y16(7),Z16(7),X17(7),Y17(7),Z17(7),X18(7),Y18(7),Z18(7),
C # X19(7),Y19(7),Z19(7),X20(7),Y20(7),Z20(7),X21(7),Y21(7),Z21(7),
C # X22(7),Y22(7),Z22(7),X23(7),Y23(7),Z23(7),X24(7),Y24(7),Z24(7),
```

```

# X25(7),Y25(7),Z25(7),X26(7),Y26(7),Z26(7),X27(7),Y27(7),Z27(7),GEN00650
# X28(7),Y28(7),Z28(7),X29(7),Y29(7),Z29(7),X30(7),Y30(7),Z30(7),GEN00660
# X31(7),Y31(7),Z31(7),X32(7),Y32(7),Z32(7),X33(7),Y33(7),Z33(7),GEN00670
# X34(7),Y34(7),Z34(7),X35(7),Y35(7),Z35(7),X36(7),Y36(7),Z36(7),GEN00680
# X37(7),Y37(7),Z37(7),X38(7),Y38(7),Z38(7),X39(7),Y39(7),Z39(7),GEN00690
# X40(7),Y40(7),Z40(7),X41(7),Y41(7),Z41(7),X42(7),Y42(7),Z42(7),GEN00700
# X43(7),Y43(7),Z43(7),X44(7),Y44(7),Z44(7),X45(7),Y45(7),Z45(7),GEN00710
# X46(7),Y46(7),Z46(7),X47(7),Y47(7),Z47(7),X48(7),Y48(7),Z48(7),GEN00720
# X49(7),Y49(7),Z49(7),X50(7),Y50(7),Z50(7),X51(7),Y51(7),Z51(7),GEN00730
# X52(7),Y52(7),Z52(7),X53(7),Y53(7),Z53(7),GEN00740

```

```

COMMON / MAIN2 / CNLTYP(100),RIBTYP(100),NPTEP(7),NPTIP(7)
# ,NPTSP(7),NPTTL,ISP,IVOLR(25),IVOLC(25),TCHNL,CCC

```

```

COMMON / SUB / ICONTI

```

```

CHARACTER STRING*40,FILE1*40,NAME*40

```

```

C *****
C *
C * THIS INTEGER STATEMENT CONVERTS THESE REAL VARI-
C * ABLES INTO INTEGERS SO THAT THEY CAN BE USED AS
C * COUNTERS
C *
C *****

```

```

INTEGER PT1,PT2,PT3,PT4,RIBTYP,CNLTYP,REGION,SDC,
# RGNCTR,SOLID,COLOR,START,STOP,COUNT,SPLINE,
# SPENTC,LNENTC,DASH,GRADE,SURFAC,ED1,ED2,ED3,ED4,
# VOLUME,SUR1,SUR2,SUR3,SUR4,SUR5,SUR6,PHCHCK,SLSLVR,
# SURT,SURB,EDGE,PNTT,PNTB,STATUS

```

```

C *****
C *
C * THIS REAL STATEMENT CONVERTS THESE INTEGER VARI-
C * ABLES INTO REALS SO THAT THEY CAN BE USED TO AS
C * ARRAYS TO HOLD THE POINT COORDINATES
C *
C *****

```

```

REAL IIX,IY,IIIX,IIY,JIX,JIY,JJIX,JJIY,KIX,KIY,
# KKIX,KKIY,LIX,LIY,LLIX,LLIY,MIX,MIY,MMIX,MMIY,
# NIX,NIY,NNIX,NNIY,LINPT

```

```

C *****
C *
C * THESE FUNCTION STATEMENTS CALCUALTE RECTANGULAR
C * POINT COORDINATES GIVEN TEH REQUIRED ANGLES
C *
C *****

```

```

XXX(Z,BETA)=Z*COS(BETA)

```

```

YYY(ZZ,BETA)=ZZ*SIN(BETA)

```

```

FNX(GAMMA)=EE*COS(3.0*GAMMA)+RR*COS(GAMMA)

```

```

FNY(GAMMA)=EE*SIN(3.0*GAMMA)+RR*SIN(GAMMA)

```

```

# FINERX(GAMMA,RLAMB,FF)=EE*COS(3.0*GAMMA)+RR*COS(GAMMA)+
# FF*COS(GAMMA+RLAMB)

```

```

# FINERY(GAMMA,RLAMB,FF)=EE*SIN(3.0*GAMMA)+RR*SIN(GAMMA)+
# FF*SIN(GAMMA+RLAMB)

```

```

GEN00750
GEN00760
GEN00770
GEN00780
GEN00790
GEN00800
GEN00810
GEN00820
GEN00830
GEN00840
GEN00850
GEN00860
GEN00870
GEN00880
GEN00890
GEN00900
GEN00910
GEN00920
GEN00930
GEN00940
GEN00950
GEN00960
GEN00970
GEN00980
GEN00990
GEN01000
GEN01010
GEN01020
GEN01030
GEN01040
GEN01050
GEN01060
GEN01070
GEN01080
GEN01090
GEN01100
GEN01110
GEN01120
GEN01130
GEN01140
GEN01150
GEN01160
GEN01170
GEN01180
GEN01190
GEN01200
GEN01210
GEN01220
GEN01230
GEN01240
GEN01250
GEN01260
GEN01270
GEN01280

```

C	*****	GEN01290
C	*	GEN01300
C	*	GEN01310
C	END FUNCTION STATEMENTS	GEN01320
C	*	GEN01330
C	*****	GEN01340
		GEN01350
C	*****	GEN01360
C	*	GEN01370
C	* CALL THE SUBROUTINE THAT CHECK FOR THE EXISTANCE	GEN01380
C	* OF THE FILE "GENERATE DATA A"	GEN01390
C	*	GEN01400
C	* IF THE FILE EXISTS, THEN READ INPUTS FOR THE PRE-	GEN01410
C	* PROCESSOR FROM THAT FILE. IF THE FILE DOES NOT	GEN01420
C	* EXIST, THEN CREATE IT AND STORE THE USER'S	GEN01430
C	* RESPONSES IN IT FOR LATER USE	GEN01440
C	*	GEN01450
C	*****	GEN01460
		GEN01470
	NO=6	GEN01480
	CALL NEWFIL (STATUS,NO)	GEN01490
		GEN01500
	IF (STATUS.EQ.0) THEN	GEN01510
		GEN01520
	PRINT*, ' '	GEN01530
	PRINT*, ' THE FILE "GENERATE DATA A" WAS FOUND ON YOUR DISK.'	GEN01540
	PRINT*, ' THEREFORE, YOU WILL NOT BE PROMPTED FOR THE INPUTS.'	GEN01550
	PRINT*, ' THE INPUTS WILL BE READ DIRECTLY FROM "GENERATE DATA".'	GEN01560
	PRINT*, ' IF YOU WANT TO BE PROMPTED FOR THE INPUTS, THEN '	GEN01570
	PRINT*, ' HALT THE EXECUTION OF THIS PROGRAM AND ERASE THE'	GEN01580
	PRINT*, ' FILE "GENERATE DATA" FROM YOUR DISK.'	GEN01590
	PRINT*, ' '	GEN01600
		GEN01610
	STRING='FILEDEF 5 DISK GENERATE DATA A '	GEN01620
	STATUS=CMSCMD (STRING)	GEN01630
		GEN01640
	IF (STATUS.NE.0) THEN	GEN01650
	PRINT*, ' FILEDEF ERROR - FILEDEF ON UNIT 5 FAILED'	GEN01660
	PRINT*, ' RETURN CODE = ',STATUS	GEN01670
	ENDIF	GEN01680
		GEN01690
	NO=7	GEN01700
		GEN01710
	NAME='FILEDEF 7 DISK STORAGE DATA A '	GEN01720
	STATUS=CMSCMD (NAME)	GEN01730
		GEN01740
	IF (STATUS.NE.0) THEN	GEN01750
	PRINT*, ' FILEDEF ERROR - FILEDEF ON UNIT 7 FAILED'	GEN01760
	PRINT*, ' RETURN CODE = ',STATUS	GEN01770
	ENDIF	GEN01780
		GEN01790
	ENDIF	GEN01800
		GEN01810
C	*****	GEN01820
C	*	GEN01830
C	*	GEN01840
C	BEGIN PROMPTED INPUT	GEN01850
C	*	GEN01860
C	*****	GEN01870
		GEN01880
C	*****	GEN01890
C	*	GEN01900
C	* ASK FOR THE HOUSING PARAMETERS	GEN01910
C	*	GEN01920
C	*****	GEN01920

```
—                                     GEN01930
WRITE(NO,*) ' '                                     GEN01940
— WRITE(NO,*) ' '                                     GEN01950
WRITE(NO,7)                                     GEN01960
7  FORMAT('0','INPUT THE ECCENTRICITY AND RADIUS OF THE ENGINE HOUSINGEN01970
#G. (EE AND RR)')                                     GEN01980
— WRITE(NO,8)                                     GEN01990
8  FORMAT('0','          OR..... ')               GEN02000
WRITE(NO,9)                                     GEN02010
— 9  FORMAT('0','INPUT THE MAX. X AND MAX. Y DIAMETER.') GEN02020
                                     GEN02030
READ(5,*) TEMP1,TEMP2                           GEN02040
IF (STATUS.NE.0) WRITE(8,*) TEMP1,TEMP2           GEN02050
WRITE(NO,*) TEMP1,TEMP2                           GEN02060
                                     GEN02070
IF (TEMP1.LT.2.0) THEN                           GEN02080
    EE=-TEMP1                                     GEN02090
    RR=TEMP2                                     GEN02100
ELSE                                              GEN02110
    RR=(TEMP1+TEMP2)/4.0                         GEN02120
    EE=-(TEMP2-TEMP1)/4.0                       GEN02130
ENDIF                                           GEN02140
    WRITE(NO,*) EE,RR                           GEN02150
                                     GEN02160
— WRITE(NO,10)                                     GEN02170
10  FORMAT('0','ENTER HOUSING THICKNESS. (D)')      GEN02180
READ(5,*) D                                     GEN02190
IF (STATUS.NE.0) WRITE(8,*) D                   GEN02200
WRITE(NO,*) D                                     GEN02210
                                     GEN02220
WRITE(NO,11)                                     GEN02230
— 11  FORMAT('0','INPUT THE DEPTH OF THE HOUSING. (DEPTH)') GEN02240
READ(5,*) DEPTH                                GEN02250
IF (STATUS.NE.0) WRITE(8,*) DEPTH                GEN02260
WRITE(NO,*) DEPTH                                GEN02270
— DEPTH=-DEPTH                                     GEN02280
                                     GEN02290
WRITE(NO,13)                                     GEN02300
13  FORMAT('0','ENTER THE THICKNESS OF THE OUTER SHELL. (AA)') GEN02310
READ(5,*) AA                                     GEN02320
IF (STATUS.NE.0) WRITE(8,*) AA                   GEN02330
WRITE(NO,*) AA                                     GEN02340
—                                     GEN02350
WRITE(NO,14)                                     GEN02360
14  FORMAT('0','ENTER THE THICKNESS OF THE INNER SHELL. (BB)') GEN02370
READ(5,*) BB                                     GEN02380
IF (STATUS.NE.0) WRITE(8,*) BB                   GEN02390
WRITE(NO,*) BB                                     GEN02400
— BB=-BB                                           GEN02410
                                     GEN02420
WRITE(NO,15)                                     GEN02430
15  FORMAT('0','ENTER THE THICKNESS OF THE INSERT. (CC)') GEN02440
WRITE(NO,16)                                     GEN02450
— 16  FORMAT('0','IF THIS MODEL IS NOT TO HAVE AN INSERT, ENTER "0.0" FOGEN02460
#R (CC).')                                     GEN02470
READ(5,*) CC                                     GEN02480
IF (STATUS.NE.0) WRITE(8,*) CC                   GEN02490
WRITE(NO,*) CC                                     GEN02500
— CCC=CC                                           GEN02510
CC=-CC                                           GEN02520
CC=BB+CC                                         GEN02530
                                     GEN02540
PI=3.14159266                                   GEN02550
                                     GEN02560
```

```
C *****
C *
C *   CALCULATE THE MAXIMUM X COORDINATE OF THE TROCHOIDAL
C *   BORE AND THE DISTANCE FROM THE CENTER OF THE CENTER
C *   HOUSING TO THE CENTER OF THE CYLINDRICAL REGIONS
C *
C *****
      ZETA=0.0
      XX(1)=RR+EE

      DO 25 JJ=2,900

      XX(JJ)=ABS(FNX(ZETA))
      IF (XX(JJ-1)-XX(JJ)) 20,20,18
18     XX(JJ)=XX(JJ-1)
20     ZETA=ZETA+0.10*PI/180.0
25     CONTINUE

      YMAX=RR-EE
      XMAX=XX(JJ-1)
      TRANS=YMAX-XMAX
      WRITE(NO,*) 'TRANS = ',TRANS
      TRANSB=TRANS
      TRANSC=TRANS
      TRANSD=TRANS
      TRANSE=TRANS
      TRANSF=TRANS
      TRANSB=TRANS

C *****
C *
C *   INITIALIZE VALUES
C *
C *****

      NN=1
      SCALE=0.1E01
      ICS=0
      ICT=0
      ICTA=0
      ITOTAL=0
      PHCHCK=0
      SLSLVR=0
      COLOR=8
      SOLID=1
      SDC=1
      ISP=0
      IL=1
      IEPQUE=0
      IIPQUE=0
      IPRTCK=0
      EPSI=0.001
      DUMMY=1.0
      SIGN=1.0
      RGNCTR=0
      ICT1=0
      ICT2=0
      ICT3=0
      ICT4=0
      R=RR-EE+D-TRANS
      T=RR-EE+D-TRANS+AA
      XMAX=RR-EE-TRANS
```

GEN02570  
GEN02580  
GEN02590  
GEN02600  
GEN02610  
GEN02620  
GEN02630  
GEN02640  
GEN02650  
GEN02660  
GEN02670  
GEN02680  
GEN02690  
GEN02700  
GEN02710  
GEN02720  
GEN02730  
GEN02740  
GEN02750  
GEN02760  
GEN02770  
GEN02780  
GEN02790  
GEN02800  
GEN02810  
GEN02820  
GEN02830  
GEN02840  
GEN02850  
GEN02860  
GEN02870  
GEN02880  
GEN02890  
GEN02900  
GEN02910  
GEN02920  
GEN02930  
GEN02940  
GEN02950  
GEN02960  
GEN02970  
GEN02980  
GEN02990  
GEN03000  
GEN03010  
GEN03020  
GEN03030  
GEN03040  
GEN03050  
GEN03060  
GEN03070  
GEN03080  
GEN03090  
GEN03100  
GEN03110  
GEN03120  
GEN03130  
GEN03140  
GEN03150  
GEN03160  
GEN03170  
GEN03180  
GEN03190  
GEN03200

```
C *****
C *
C *          END INITIALIZTION OF VALUES
C *
C *****
29      WRITE(13,29) PI,R,TRANS
      FORM 7 (F13.5,2X,F13.5,2X,F13.5)
      DO 130 J=1,200
          IT=1
          IVOLC(J)=0
          IVOLR(J)=0
C *****
C *
C *          ASK FOR THE REGION TYPE
C *
C *****
30      WRITE(NO,30)
      FORMAT('0',' IN WHICH REGION DOES THE RIB-CHANNEL SEGMENT LIE?')
      WRITE(NO,32)
32      FORMAT('0',' ENTER 1 FOR PHI-SOLVER REGION...')
      WRITE(NO,33)
33      FORMAT('0',' ENTER 2 FOR STRAIGHT-LINE SOLVER REGION...')
      WRITE(NO,34)
34      FORMAT('0',' ENTER 3 IF THE RC SEGMENT OVERLAPS THE REGIONS...
      #')
      WRITE(NO,36)
36      FORMAT('0',' OR ENTER 4 TO END INPUT.')
          READ(5,*) REGION(J)
          IF (STATUS.NE.0) WRITE(8,*) REGION(J)
          WRITE(NO,*) REGION(J)
          IF(REGION(J).EQ.4) GO TO 135
          WRITE(13,*) REGION(J)
          IF(REGION(J).EQ.3) THEN
              RGNCTR=RGNCTR+1
              IF(RGNCTR.EQ.1) THEN
                  ICT1=J
              ELSE IF(RGNCTR.EQ.2) THEN
                  ICT2=J
              ELSE IF (RGNCTR.EQ.3) THEN
                  ICT3=J
              ELSE IF (RGNCTR.EQ.4) THEN
                  ICT4=J
              ENDIF
          ENDIF
C *****
C *
C *          IF THE REGION IS CYLINDRICAL, THEN ASK FOR THE TWO
C *          ANGLES THAT DEFINE THE LEADING AND TRAILING EDGE
C *          OF THE RIB
C *
```

```
GEN03210
GEN03220
GEN03230
GEN03240
GEN03250
GEN03260
GEN03270
GEN03280
GEN03290
GEN03300
GEN03310
GEN03320
GEN03330
GEN03340
GEN03350
GEN03360
GEN03370
GEN03380
GEN03390
GEN03400
GEN03410
GEN03420
GEN03430
GEN03440
GEN03450
GEN03460
GEN03470
GEN03480
GEN03490
GEN03500
GEN03510
GEN03520
GEN03530
GEN03540
GEN03550
GEN03560
GEN03570
GEN03580
GEN03590
GEN03600
GEN03610
GEN03620
GEN03630
GEN03640
GEN03650
GEN03660
GEN03670
GEN03680
GEN03690
GEN03700
GEN03710
GEN03720
GEN03730
GEN03740
GEN03750
GEN03760
GEN03770
GEN03780
GEN03790
GEN03800
GEN03810
GEN03820
GEN03830
GEN03840
```



```
C *****
C                                     GEN03850
C                                     GEN03860
C      PHI1OD=PHIONE                                     GEN03870
C      PHI2OD=PHITWO                                     GEN03880
C      IF (REGION(J).EQ.1) THEN                           GEN03890
C          WRITE(NO,40)                                    GEN03900
40      FORMAT('0','ENTER A VALUE FOR PHIONE AND PHITWO (DEGREES).') GEN03910
C          READ(5,*) PHIONE,PHITWO                        GEN03920
C          IF (STATUS.NE.0) WRITE(8,*) PHIONE,PHITWO      GEN03930
C          WRITE(NO,*) PHIONE,PHITWO                      GEN03940
C          WRITE(13,42) PHIONE,PHITWO                    GEN03950
42      FORMAT(F13.5,5X,F13.5)                             GEN03960
C          PHI1(J)=PHIONE                                 GEN03970
C                                                         GEN03980
C                                                         GEN03990
C      *                                                  GEN04000
C      *          IF THE REGION IS RECTANGULAR, THEN ASK FOR THE TWO *      GEN04010
C      *          Y-COORDINATES THAT DEFINE THE LEADING ANB TRAILING *      GEN04020
C      *          EDGE OF THE RIB                             *      GEN04030
C      *                                                  *      GEN04040
C      *                                                  *      GEN04050
C      *****                                           GEN04060
C          ELSE IF (REGION(J).EQ.2) THEN                   GEN04070
C              WRITE(NO,44)                                GEN04080
44      FORMAT('0','ENTER THE YONE AND Y TWO COORDINATE OF THE RIB CEN GEN04090
C      #TER.')                                           GEN04100
C              READ(5,*) YONE,YTWO                         GEN04110
C              IF (STATUS.NE.0) WRITE(8,*) YONE,YTWO      GEN04120
C              WRITE(NO,*) YONE,YTWO                      GEN04130
C              WRITE(13,42) YONE,YTWO                    GEN04140
C              YI=YONE                                     GEN04150
C                                                         GEN04160
C      *****                                           GEN04170
C      *                                                  *      GEN04180
C      *          IF THE REGION IS OVERLAPPING, THEN ASK FOR THE TWO *      GEN04190
C      *          ANGLES THAT DEFINE THE LEADING AND TRAILING EDGE *      GEN04200
C      *          OF THE RIB                             *      GEN04210
C      *                                                  *      GEN04220
C      *                                                  *      GEN04230
C      *****                                           GEN04240
C          ELSE IF (REGION(J).EQ.3) THEN                   GEN04250
C                                                         GEN04260
C              WRITE(NO,55)                                GEN04270
55      FORMAT('0','ENTER A VALUE FOR PHIONE AND PHITWO (DEGREES).') GEN04280
C              READ(5,*) PHIONE,PHITWO                    GEN04290
C              IF (STATUS.NE.0) WRITE(8,*) PHIONE,PHITWO  GEN04300
C              WRITE(NO,*) PHIONE,PHITWO                  GEN04310
C              WRITE(13,42) PHIONE,PHITWO                 GEN04320
C              PHI1(J)=PHIONE                             GEN04330
C          ENDIF                                           GEN04340
C                                                         GEN04350
C      *****                                           GEN04360
C      *                                                  *      GEN04370
C      *          ASK FOR THE RIB TYPE                     *      GEN04380
C      *                                                  *      GEN04390
C      *                                                  *      GEN04400
C      *****                                           GEN04410
C          WRITE(NO,60)                                    GEN04420
60      FORMAT('0','WHAT TYPE OF RIB IS TO BE INPUT?')   GEN04430
C          WRITE(NO,62)                                    GEN04440
62      FORMAT('0','    ENTER 1 FOR NORMAL RIB...')      GEN04450
C          WRITE(NO,64)                                    GEN04460
64      FORMAT('0','    ENTER 2 FOR RECESSED RIB...')    GEN04470
C          WRITE(NO,66)                                    GEN04480
```

```
66      FORMAT('0','OR ENTER 3 IF THIS RIB IS A SPARK PLUG.')      GEN04490
      READ(5,*) RIBTYP(J)      GEN04500
      IF (STATUS.NE.0) WRITE(8,*) RIBTYP(J)      GEN04510
      WRITE(NO,*) RIBTYP(J)      GEN04520
      WRITE(13,43) RIBTYP(J)      GEN04530
43      FORMAT(I2)      GEN04540
      GEN04550
C      *****      GEN04560
C      *      GEN04570
C      *      IF THE RIB IS RECESSED, ASK FOR MAGNITUDE OF RECESS      *      GEN04580
C      *      *      GEN04590
C      *****      GEN04600
      IF(RIBTYP(J).EQ.2) THEN      GEN04610
      WRITE(NO,68)      GEN04620
68      FORMAT('0','ENTER THE MAGNITUDE OF THE RECESS.')      GEN04630
      READ(5,*) RECES      GEN04640
      IF (STATUS.NE.0) WRITE(8,*) RECES      GEN04650
      WRITE(NO,*) RECES      GEN04660
      GEN04670
      GEN04680
C      *****      GEN04690
C      *      GEN04700
C      *      IF THE RIB IS A SPARK PLUG PORT, ASK FOR THE RADIUS      *      GEN04710
C      *      OF THE PORT      *      GEN04720
C      *      *      GEN04730
C      *****      GEN04740
      ELSE IF (RIBTYP(J).EQ.3) THEN      GEN04750
      WRITE(NO,69)      GEN04760
69      FORMAT('0','ENTER THE RADIUS OF THE SPARK PLUG.')      GEN04770
      READ(5,*) RSPPG      GEN04780
      IF (STATUS.NE.0) WRITE(8,*) RSPPG      GEN04790
      WRITE(NO,*) RSPPG      GEN04800
      RIBTYP(J)=1      GEN04810
      IVOLR(J)=J      GEN04820
      GEN04830
      ISP=ISP+1      GEN04840
      RSP(ISP)=RSPPG      GEN04850
      PHI1SP(ISP)=PHIONE*PI/180.0      GEN04860
      PHI2SP(ISP)=PHITWO*PI/180.0      GEN04870
      Y1SP(ISP)=YONE      GEN04880
      Y2SP(ISP)=YTWO      GEN04890
      ISPRK(ISP)=J      GEN04900
      GEN04910
      ENDIF      GEN04920
      GEN04930
      GEN04940
      GEN04950
C      *****      GEN04960
C      *      GEN04970
C      *      ASK FOR THE CHANNEL TYPE      *      GEN04980
C      *      *      GEN04990
C      *****      GEN05000
      IF((J.EQ.1).OR.((RIBTYP(J).EQ.1).AND.(RIBTYP(J-1).EQ.1))) THEN      GEN05010
      WRITE(NO,70)      GEN05020
70      FORMAT('0','WHAT TYPE OF CHANNEL IS TO BE INPUT?')      GEN05030
      WRITE(NO,72)      GEN05040
72      FORMAT('0','ENTER 1 FOR NORMAL CHANNEL...')      GEN05050
      WRITE(NO,74)      GEN05060
74      FORMAT('0','ENTER 2 FOR STIFFENED CHANNEL ...')      GEN05070
      WRITE(NO,75)      GEN05080
75      FORMAT('0','ENTER 3 IF THIS CHANNEL IS THE EXHAUST PORT ...')      GEN05090
      WRITE(NO,78)      GEN05100
      GEN05110
      GEN05120
```

```
78  FORMAT('0','OR ENTER 4 IF THIS CHANNEL IS THE INTAKE PORT.')
      READ(5,*) CNLTYP(J)
      IF (STATUS.NE.0) WRITE(8,*) CNLTYP(J)
      WRITE(NO,*) CNLTYP(J)

      IF(IEPQUE.EQ.1) GO TO 84

C *****
C *
C * IF THE CHANNEL IS AN EXHAUST PORT, THEN ASK FOR THE
C * TWO ANGLES THAT DEFINE THE SIZE AND LOCATION OF
C * THE PORT
C *
C *****
      IF(CNLTYP(J).EQ.3) THEN
        IEPQUE=1
        IVOLC(J)=J
        CNLTYP(J)=1
        IEP=J
        P1ODEP=2.0*PI-PHI1OD
        P2ODEP=2.0*PI-PHI2OD
        P1NWEF=2.0*PI-(PHIONE*PI/180.0)

      WRITE(NO,79)
79  FORMAT('0','ENTER PHIONE AND PHITWO OF THE EXHAUST PORT.')
      READ(5,*) PHI1EP,PHI2EP
      IF (STATUS.NE.0) WRITE(8,*) PHI1EP,PHI2EP
      PHI1EP=2.0*PI-(PHI1EP*PI/180.0)
      PHI2EP=2.0*PI-(PHI2EP*PI/180.0)
      WRITE(NO,*) PHI1EP,PHI2EP

      WRITE(NO,80)
80  FORMAT('0','ENTER THE RADIUS OF THE EXHAUST PORT.')
      READ(5,*) REXPT
      IF (STATUS.NE.0) WRITE(8,*) REXPT
      WRITE(NO,*) REXPT

      WRITE(NO,82)
82  FORMAT('0','ENTER THE THICKNESS OF THE EXHAUST PORT.')
      READ(5,*) TEXPT
      IF (STATUS.NE.0) WRITE(8,*) TEXPT
      WRITE(NO,*) TEXPT

      ENDIF

C *****
C *
C * IF THE CHANNEL IS AN INTAKE PORT, THEN ASK FOR THE
C * TWO ANGLES THAT DEFINE THE SIZE AND LOCATION OF
C * THE PORT
C *
C *****
84  IF(IIPQUE.EQ.1) GO TO 96

      IF(CNLTYP(J).EQ.4) THEN
        IIPQUE=1
        IVOLC(J)=J
        CNLTYP(J)=1
        IIP=J
        P2ODIP=PHI2OD
        P1NWIP=PHIONE*PI/180.0
```

```
      PHIZ=PINWIP*180.0/PI
      WRITE(NO,86)
86  FORMAT('0','ENTER PHIONE AND PHITWO OF THE INTAKE PORT.')
      READ(5,*) PH1IP,PHI2IP
      IF (STATUS.NE.0) WRITE(8,*) PH1IP,PHI2IP
      PH1IP=PH1IP*PI/180.0
      PHI2IP=PHI2IP*PI/180.0
      WRITE(NO,*) PH1IP,PHI2IP

      WRITE(NO,88)
88  FORMAT('0','ENTER THE RADIUS OF THE INTAKE PORT.')
      READ(5,*) RINPT
      IF (STATUS.NE.0) WRITE(8,*) RINPT
      WRITE(NO,*) RINPT

      WRITE(NO,89)
89  FORMAT('0','ENTER THE THICKNESS OF THE INTAKE PORT.')
      READ(5,*) TINPT
      IF (STATUS.NE.0) WRITE(8,*) TINPT
      WRITE(NO,*) TINPT

      WRITE(NO,91)
91  FORMAT('0','ENTER THE LENGTH AND WIDTH OF THE INTAKE PORT.')
      READ(5,*) LINPT,WINPT
      IF (STATUS.NE.0) WRITE(8,*) LINPT,WINPT
      WRITE(NO,*) LINPT,WINPT

      ENDIF

96  CONTINUE
      ELSE

      WRITE(NO,98)
98  FORMAT('0','THE CHANNEL MUST BE NORMAL.')
      CNLTYP(J)=1

      ENDIF

      WRITE(13,*) IVOLC(J),IVOLR(J)

C *****
C *
C *
C *
C *
C *****
      IF((PHIONE.GT.180.0).AND.(PHIONE.LT.270.0)) PHCHCK=PHCHCK+1
      IF((PHITWO.GT.180.0).AND.(PHITWO.LT.270.0)) PHCHCK=PHCHCK+1
      IF(PHIONE.GT.360.0) PHCHCK=PHCHCK+1
      IF(PHITWO.GT.360.0) PHCHCK=PHCHCK+1

      PHIONE=PHIONE*PI/180.0
      PHITWO=PHITWO*PI/180.0

C *****
C *
C *
C *
C *
C *
C *
C *****
      DEPENDING UPON WHICH REGION WAS SPECIFIED BY THE
      USER, THE APPROPRIATE SECTION OF THE PROGRAM IS
      CALLED
      EITHER THE OVERLAP, RECTANGULAR OR CYLINDRICAL
      *****
```

```
IF(REGION(J).EQ.1) THEN
```

```
GO TO 120
```

```
ELSE IF(REGION(J).EQ.2) THEN
```

```
GO TO 115
```

```
ELSE IF((RGNCTR.EQ.1).OR.(RGNCTR.EQ.3)) THEN
```

```
GO TO 100
```

```
ELSE IF((RGNCTR.EQ.2).OR.(RGNCTR.EQ.4)) THEN
```

```
GO TO 105
```

```
ENDIF
```

```
C *****
C *
C * THIS SECTION CALCULATES THE POINT COORDINATES OF THE *
C * OVERLAPPING RIB/CHANNEL SEGMENTS. THERE ARE 12 *
C * DIFFERENT OVERLAP CONDITIONS DEPENDING UPON THE *
C * PERCENT A GIVEN RIB OR CHANNEL OVERLAPS A REGION *
C *
C *****
```

```
100 IF(J.EQ.1) THEN
```

```
C *****
C *
C * FOR THE FIRST RIB/CAHNNEL SEGMENT, ASK FOR THE *
C * THICKNESS OF THE CHANNEL *
C *
C *****
```

```
WRITE(NO,102)
```

```
102 FORMAT('0','ENTER THE THICKNESS OF THE CHANNEL.')
```

```
READ(5,*) TCHNL
```

```
IF (STATUS.NE.0) WRITE(8,*) TCHNL
```

```
WRITE(NO,*) TCHNL
```

```
ENDIF
```

```
IF((PHIONE.LT.0.0).OR.((SLSLVR.GT.1).AND.(PHIONE.LT.PI))) THEN
```

```
IF(SLSLVR.GT.1) THEN
```

```
SIGN=-SIGN
```

```
TRANS=-TRANS
```

```
RECES=-RECES
```

```
ENDIF
```

```
ICT=ICT+1
```

```
IF(ICT.EQ.1) ICHK=0
```

```
IF(ICT.GT.1) ICHK=1
```

```
PHIONE=-PHIONE
```

```
VRTICL=T*TAN(PHIONE)
```

```
KKIX(J)=T*SIGN
```

```
KKIY(J)=(ABS(TRANS)-VRTICL)*SIGN
```

```
KIX(J)=R*SIGN
```

```
KIY(J)=KKIY(J)
```

GEN06410  
GEN06420  
GEN06430  
GEN06440  
GEN06450  
GEN06460  
GEN06470  
GEN06480  
GEN06490  
GEN06500  
GEN06510  
GEN06520  
GEN06530  
GEN06540  
GEN06550  
GEN06560  
GEN06570  
GEN06580  
GEN06590  
GEN06600  
GEN06610  
GEN06620  
GEN06630  
GEN06640  
GEN06650  
GEN06660  
GEN06670  
GEN06680  
GEN06690  
GEN06700  
GEN06710  
GEN06720  
GEN06730  
GEN06740  
GEN06750  
GEN06760  
GEN06770  
GEN06780  
GEN06790  
GEN06800  
GEN06810  
GEN06820  
GEN06830  
GEN06840  
GEN06850  
GEN06860  
GEN06870  
GEN06880  
GEN06890  
GEN06900  
GEN06910  
GEN06920  
GEN06930  
GEN06940  
GEN06950  
GEN06960  
GEN06970  
GEN06980  
GEN06990  
GEN07000  
GEN07010  
GEN07020  
GEN07030  
GEN07040

```

      DIX(J)=XMAX*SIGN
      DIY(J)=KKIY(J)
      IF(RIBTYP(J).EQ.2) THEN

          DRIX(J)=DIX(J)+RECES
          DRIY(J)=DIY(J)+RECES

      ENDIF

      CALL YALPSL (EE,RR,DIY(J),ICLK,DALPHA)
      CALL THTASL (EE,RR,PI,DALPHA,DTHETA)

      NNIX(J)=XXX(T,PHITWO)
      NNIY(J)=(ABS(YYY(T,PHITWO))+ABS(TRANS))*SIGN
      NIX(J)=XXX(R,PHITWO)
      NIY(J)=(ABS(YYY(R,PHITWO))+ABS(TRANS))*SIGN
      PHIG=PHITWO

      CALL PALPSL (EE,RR,PI,PHITWO,TRANS,GALPHA)
      CALL THTASL (EE,RR,PI,GALPHA,GTHETA)

      GIX(J)=FNX(GALPHA)
      GIY(J)=FNY(GALPHA)

      IF(RIBTYP(J).EQ.2) THEN

          GRIX(J)=GIX(J)+RECES*COS(PHIG)
          GRIY(J)=GIY(J)+RECES*SIN(PHIG)

      ENDIF

      CHECK=ABS(NNIY(J))-ABS(TRANS)+ABS(VRTICL)

C *****
C *
C *          OVERLAP CONDITION ONE
C *
C *      20% OF THE RIB LIES IN THE CYLINDRICAL REGION
C *      (QUADRANTS ONE AND THREE)
C *
C *****

      IF(VRTICL.LT.CHECK/5.0) THEN
      WRITE(NO,*) 'PART 1'
      IF(PHITWO.LT.PI) THEN

          PHIE=PHITWO/3.0
          PHIF=2.0*PHITWO/3.0

      ELSE IF(PHITWO.GT.PI) THEN

          PHIE=PI+(PHITWO-PI)/3.0
          PHIF=PI+2.0*(PHITWO-PI)/3.0

      ENDIF

      LLIX(J)=XXX(T,PHIE)
      LLIY(J)=YYY(T,PHIE)+TRANS
      LIX(J)=XXX(R,PHIE)
      LIY(J)=YYY(R,PHIE)+TRANS

      CALL PALPSL (EE,RR,PI,PHIE,TRANS,EALPHA)
      CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
```

GEN07050  
GEN07060  
GEN07070  
GEN07080  
GEN07090  
GEN07100  
GEN07110  
GEN07120  
GEN07130  
GEN07140  
GEN07150  
GEN07160  
GEN07170  
GEN07180  
GEN07190  
GEN07200  
GEN07210  
GEN07220  
GEN07230  
GEN07240  
GEN07250  
GEN07260  
GEN07270  
GEN07280  
GEN07290  
GEN07300  
GEN07310  
GEN07320  
GEN07330  
GEN07340  
GEN07350  
GEN07360  
GEN07370  
GEN07380  
GEN07390  
GEN07400  
GEN07410  
GEN07420  
GEN07430  
GEN07440  
GEN07450  
GEN07460  
GEN07470  
GEN07480  
GEN07490  
GEN07500  
GEN07510  
GEN07520  
GEN07530  
GEN07540  
GEN07550  
GEN07560  
GEN07570  
GEN07580  
GEN07590  
GEN07600  
GEN07610  
GEN07620  
GEN07630  
GEN07640  
GEN07650  
GEN07660  
GEN07670  
GEN07680

```
      EIX(J)=FNX(EALPHA)
      EIY(J)=FNY(EALPHA)

      IF(RIBTYP(J).EQ.2) THEN

        ERIX(J)=EIX(J)+RECES*COS(PHIE)
        ERIY(J)=EIY(J)+RECES*SIN(PHIE)

      ENDIF

      MMIX(J)=XXX(T,PHIF)
      MMIY(J)=YYY(T,PHIF)+TRANS
      MIX(J)=XXX(R,PHIF)
      MIY(J)=YYY(R,PHIF)+TRANS

      CALL PALPSL (EE,RR,PI,PHIF,TRANS,FALPHA)
      CALL THTASL (EE,RR,PI,FALPHA,FTHETA)

      FIX(J)=FNX(FALPHA)
      FIY(J)=FNY(FALPHA)

      IF(RIBTYP(J).EQ.2) THEN

        FRIX(J)=FIX(J)+RECES*COS(PHIF)
        FRIY(J)=FIY(J)+RECES*SIN(PHIF)

      ENDIF

      *****
      *
      *          OVERLAP CONDITION TWO
      *
      *      80% OF THE RIB LIES IN THE CYLINDRICAL REGION
      *          (QUADRANTS ONE AND THREE)
      *
      *****

      ELSE IF (VRTICL.GT.4.0*CHECK/5.0) THEN
        WRITE(NO,*) 'PART 2'
        LLIJ(J)=T*SIGN
        LLIY(J)=(ABS(KKIY(J))+VRTICL/3.0)*SIGN
        LIJ(J)=R*SIGN
        LIY(J)=LLIJ(J)
        EIX(J)=XMAX*SIGN
        EIY(J)=LLIY(J)

        IF (RIBTYP(J).EQ.2) THEN

          ERIX(J)=EIX(J)+RECES
          ERIY(J)=EIY(J)+RECES

        ENDIF

        CALL YALPSL (EE,RR,LIY(J),ICLK,EALPHA)
        CALL THTASL (EE,RR,PI,EALPHA,ETHETA)

        MMIX(J)=T*SIGN
        MMIY(J)=(ABS(KKIY(J))+2.0*VRTICL/3.0)*SIGN
        MIX(J)=R*SIGN
        MIY(J)=MMIY(J)
        FIX(J)=XMAX*SIGN
        FIY(J)=MMIY(J)
```

```
GEN07690
GEN07700
GEN07710
GEN07720
GEN07730
GEN07740
GEN07750
GEN07760
GEN07770
GEN07780
GEN07790
GEN07800
GEN07810
GEN07820
GEN07830
GEN07840
GEN07850
GEN07860
GEN07870
GEN07880
GEN07890
GEN07900
GEN07910
GEN07920
GEN07930
GEN07940
GEN07950
GEN07960
GEN07970
GEN07980
GEN07990
GEN08000
GEN08010
GEN08020
GEN08030
GEN08040
GEN08050
GEN08060
GEN08070
GEN08080
GEN08090
GEN08100
GEN08110
GEN08120
GEN08130
GEN08140
GEN08150
GEN08160
GEN08170
GEN08180
GEN08190
GEN08200
GEN08210
GEN08220
GEN08230
GEN08240
GEN08250
GEN08260
GEN08270
GEN08280
GEN08290
GEN08300
GEN08310
GEN08320
```

```
IF (RIBTYP(J).EQ.2) THEN
    FRIX(J)=FIX(J)+RECES
    FRIY(J)=FIY(J)+RECES
ENDIF

CALL YALPSL (EE,RR,MIY(J),ICLK,FALPHA)
CALL THTASL (EE,RR,PI,FALPHA,FTHETA)

*****
*
*          OVERLAP CONDITION THREE
*
*   BETWEEN 20% AND 80% OF THE RIB LIES IN THE
*   CYLINDRICAL REGION
*   (QUADRANTS ONE AND THREE)
*
*****

ELSE IF ((VRTICL.GT.CHECK/5.) .AND. (VRTICL.LT.4.*CHECK/5.)) THEN
    WRITE(NO,*) 'PART 3'
    LLIX(J)=T*SIGN
    LLIY(J)=(ABS(KKIY(J))+VRTICL/2.0)*SIGN
    LIX(J)=R*SIGN
    LIY(J)=LLIY(J)
    EIX(J)=XMAX*SIGN
    EIY(J)=LLIY(J)

    IF (RIBTYP(J).EQ.2) THEN
        ERIX(J)=EIX(J)+RECES
        ERIY(J)=EIY(J)+RECES
    ENDIF

    CALL YALPSL (EE,RR,EIY(J),ICLK,EALPHA)
    CALL THTASL (EE,RR,PI,EALPHA,ETHETA)

    IF (PHITWO.LT.PI) THEN
        PHIF=PHITWO/2.0
    ELSE IF (PHITWO.GT.PI) THEN
        PHIF=PI+(PHITWO-PI)/2.0
    ENDIF

    MMIX(J)=XXX(T,PHIF)
    MMIY(J)=(YYY(T,PHIF)+TRANS)*SIGN
    MIX(J)=XXX(R,PHIF)
    MIY(J)=(YYY(R,PHIF)+TRANS)*SIGN

    CALL PALPSL (EE,RR,PI,PHIF,TRANS,FALPHA)
    CALL THTASL (EE,RR,PI,FALPHA,FTHETA)

    FIX(J)=FNX(FALPHA)
    FIY(J)=FNY(FALPHA)*SIGN

    IF (RIBTYP(J).EQ.2) THEN
        FRIX(J)=FIX(J)+RECES*COS(PHIF)
        FRIY(J)=FIY(J)+RECES*SIN(PHIF)
    ENDIF
```



```

      ENDIF
      IF(J.EQ.1) THEN
        CHECK=TCHNL
      ELSE
        CHECK=KKIY(J)-NNIY(J-1)
      ENDIF
      *****
      *
      *   THIS SECTION CALCUALTES THE POINT COORDINATES OF THE
      *   CHANNEL FOR THE FIRST THREE OVERLAP CONDITIONS
      *
      *****
      IIX(J)=T*SIGN
      IF(J.EQ.1) THEN
        IIIY(J)=KKIY(J)-2.0*TCHNL/3.0
        AIX(J)=XMAX*SIGN
        AIY(J)=(KKIY(J)-TCHNL)*SIGN
      CALL YALPSL (EE,RR,AIY(J),ICLK,AALPHA)
      CALL THTASL (EE,RR,PI,AALPHA,ATHETA)
      AAAIX(J)=FINERX(AALPHA,ATHETA,CC)
      AAAIY(J)=FINERY(AALPHA,ATHETA,CC)
      AAIX(J)=FINERX(AALPHA,ATHETA,BB)
      AAIY(J)=FINERY(AALPHA,ATHETA,BB)
      ELSE
        IIIY(J)=(ABS(KKIY(J))-ABS(2.0*(KKIY(J)-NNIY(J-1))/3.0))*SIGN
      ENDIF
      IIX(J)=R*SIGN
      IIY(J)=IIIY(J)
      BIX(J)=XMAX*SIGN
      BIY(J)=IIIY(J)
      CALL YALPSL (EE,RR,BIY(J),ICLK,BALPHA)
      CALL THTASL (EE,RR,PI,BALPHA,BTHETA)
      JJIX(J)=T*SIGN
      IF(J.EQ.1) THEN
        JJIY(J)=KKIY(J)-TCHNL/3.0
      ELSE
        JJIY(J)=(ABS(KKIY(J))-ABS((KKIY(J)-NNIY(J-1))/3.0))*SIGN
      ENDIF
      JIX(J)=R*SIGN
      JIY(J)=JJY(J)
      CIX(J)=XMAX*SIGN
      CIY(J)=JJY(J)
      CALL YALPSL (EE,RR,CIY(J),ICLK,CALPHA)
      CALL THTASL (EE,RR,PI,CALPHA,CTHETA)

```

```

GEN08970
GEN08980
GEN08990
GEN09000
GEN09010
GEN09020
GEN09030
GEN09040
GEN09050
GEN09060
GEN09070
GEN09080
GEN09090
GEN09100
GEN09110
GEN09120
GEN09130
GEN09140
GEN09150
GEN09160
GEN09170
GEN09180
GEN09190
GEN09200
GEN09210
GEN09220
GEN09230
GEN09240
GEN09250
GEN09260
GEN09270
GEN09280
GEN09290
GEN09300
GEN09310
GEN09320
GEN09330
GEN09340
GEN09350
GEN09360
GEN09370
GEN09380
GEN09390
GEN09400
GEN09410
GEN09420
GEN09430
GEN09440
GEN09450
GEN09460
GEN09470
GEN09480
GEN09490
GEN09500
GEN09510
GEN09520
GEN09530
GEN09540
GEN09550
GEN09560
GEN09570
GEN09580
GEN09590
GEN09600

```

```
HIX(J)=R*SIGN
HIY(J)=(KKIY(J)-CHECK)*SIGN
```

```
SIGN=ABS(SIGN)
TRANS=ABS(TRANS)
RECES=ABS(RECES)
```

```
*****
*
*      END CHANNEL COORDINATE CALCULATION
*      OF FIRST THREE OVERLAP CONDITIONS
*
*****
```

```
ELSE IF (PHIONE.GE.0.0) THEN
  IF (SLSLVR.GT.1) THEN
```

```
    SIGN=-SIGN
    TRANS=-TRANS
    RECES=-RECES
```

```
  ENDIF
```

```
  PHIE=PHIONE+(PHITWO-PHIONE)/3.0
  PHIF=PHIONE+2.0*(PHITWO-PHIONE)/3.0
```

```
    KKIX(J)=XXX(T,PHIONE)
    KKIY(J)=YYY(T,PHIONE)+TRANS
    KIX(J)=XXX(R,PHIONE)
    KIY(J)=YYY(R,PHIONE)+TRANS
    PHID=PHIONE
```

```
    CALL PALPSL (EE,RR,PI,PHIONE,TRANS,DALPHA)
    CALL THTASL (EE,RR,PI,DALPHA,DTHETA)
```

```
    DIX(J)=FNX(DALPHA)
    DIY(J)=FNY(DALPHA)
```

```
  IF (RIBTYP(J).EQ.2) THEN
```

```
    DRIX(J)=DIX(J)+RECES*COS(PHID)
    DRIY(J)=DIY(J)+RECES*SIN(PHID)
```

```
  ENDIF
```

```
    LLIJ(J)=XXX(T,PHIE)
    LLIJ(J)=YYY(T,PHIE)+TRANS
    LIJ(J)=XXX(R,PHIE)
    LIY(J)=YYY(R,PHIE)+TRANS
```

```
    CALL PALPSL (EE,RR,PI,PHIE,TRANS,EALPHA)
    CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
```

```
    EIX(J)=FNX(EALPHA)
    EIJ(J)=FNY(EALPHA)
```

```
  IF (RIBTYP(J).EQ.2) THEN
```

```
    ERIX(J)=EIX(J)+RECES*COS(PHIE)
    ERIY(J)=EIJ(J)+RECES*SIN(PHIE)
```

```
  ENDIF
```

GEN09610  
GEN09620  
GEN09630  
GEN09640  
GEN09650  
GEN09660  
GEN09670  
GEN09680  
GEN09690  
GEN09700  
GEN09710  
GEN09720  
GEN09730  
GEN09740  
GEN09750  
GEN09760  
GEN09770  
GEN09780  
GEN09790  
GEN09800  
GEN09810  
GEN09820  
GEN09830  
GEN09840  
GEN09850  
GEN09860  
GEN09870  
GEN09880  
GEN09890  
GEN09900  
GEN09910  
GEN09920  
GEN09930  
GEN09940  
GEN09950  
GEN09960  
GEN09970  
GEN09980  
GEN09990  
GEN10000  
GEN10010  
GEN10020  
GEN10030  
GEN10040  
GEN10050  
GEN10060  
GEN10070  
GEN10080  
GEN10090  
GEN10100  
GEN10110  
GEN10120  
GEN10130  
GEN10140  
GEN10150  
GEN10160  
GEN10170  
GEN10180  
GEN10190  
GEN10200  
GEN10210  
GEN10220  
GEN10230  
GEN10240

```
MMIX(J)=XXX(T,PHIF)
MMIY(J)=YYY(T,PHIF)+TRANS
MIX(J)=XXX(R,PHIF)
MIY(J)=YYY(R,PHIF)+TRANS

CALL PALPSL (EE,RR,PI,PHIF,TRANS,FALPHA)
CALL THTASL (EE,RR,PI,FALPHA,FTHETA)

FIX(J)=FNX(FALPHA)
FIY(J)=FNY(FALPHA)

IF(RIBTYP(J).EQ.2) THEN

    FRIX(J)=FIX(J)+RECES*COS(PHIF)
    FRIY(J)=FIY(J)+RECES*SIN(PHIF)

ENDIF

NNIX(J)=XXX(T,PHITWO)
NNIY(J)=YYY(T,PHITWO)+TRANS
NIX(J)=XXX(R,PHITWO)
NIY(J)=YYY(R,PHITWO)+TRANS
PHIG=PHITWO

CALL PALPSL (EE,RR,PI,PHITWO,TRANS,GALPHA)
CALL THTASL (EE,RR,PI,GALPHA,GTHETA)

GIX(J)=FNX(GALPHA)
GIY(J)=FNY(GALPHA)

IF(RIBTYP(J).EQ.2) THEN

    GRIX(J)=GIX(J)+RECES*COS(PHIG)
    GRIY(J)=GIY(J)+RECES*SIN(PHIG)

ENDIF

IF(J.EQ.1) THEN

    AIX(J)=XMAX*SIGN
    AIY(J)=(KKIY(J)-TCHNL)*SIGN

    CALL YALPSL (EE,RR,AIY(J),ICLK,AALPHA)
    CALL THTASL (EE,RR,PI,AALPHA,ATHETA)

    AAAIX(J)=FINERX(AALPHA,ATHETA,CC)
    AAAIY(J)=FINERY(AALPHA,ATHETA,CC)
    AAIX(J)=FINERX(AALPHA,ATHETA,BB)
    AAIY(J)=FINERY(AALPHA,ATHETA,BB)

ENDIF

IF(J.EQ.1) THEN
    CHECK=TCHNL
ELSE
    CHECK=ABS(KKIY(J))-ABS(NNIY(J-1))
    CHECK=ABS(CHECK)
ENDIF

HIX(J)=R*SIGN
```

GEN10250  
GEN10260  
GEN10270  
GEN10280  
GEN10290  
GEN10300  
GEN10310  
GEN10320  
GEN10330  
GEN10340  
GEN10350  
GEN10360  
GEN10370  
GEN10380  
GEN10390  
GEN10400  
GEN10410  
GEN10420  
GEN10430  
GEN10440  
GEN10450  
GEN10460  
GEN10470  
GEN10480  
GEN10490  
GEN10500  
GEN10510  
GEN10520  
GEN10530  
GEN10540  
GEN10550  
GEN10560  
GEN10570  
GEN10580  
GEN10590  
GEN10600  
GEN10610  
GEN10620  
GEN10630  
GEN10640  
GEN10650  
GEN10660  
GEN10670  
GEN10680  
GEN10690  
GEN10700  
GEN10710  
GEN10720  
GEN10730  
GEN10740  
GEN10750  
GEN10760  
GEN10770  
GEN10780  
GEN10790  
GEN10800  
GEN10810  
GEN10820  
GEN10830  
GEN10840  
GEN10850  
GEN10860  
GEN10870  
GEN10880

HIY(J)=(KKIY(J)-CHECK)\*SIGN

VRTICL=ABS(KKIY(J))-ABS(TRANS)

```
C *****
C *
C *          OVERLAP CONDITION FOUR
C *
C *    20% OF THE RIB LIES IN THE CYLINDRICAL REGION
C *    (QUADRANTS ONE AND THREE)
C *
C *****
```

```
IF(VRTICL.LT.CHECK/5.0) THEN
  WRITE(NO,*) 'PART 4'
  IIIX(J)=T*SIGN
  IIIY(J)=(ABS(KKIY(J))-2.0*CHECK/3.0)*SIGN
  IIX(J)=R*SIGN
  IIY(J)=IIY(J)
  BIX(J)=XMAX*SIGN
  BIY(J)=IIY(J)
```

```
CALL YALPSL (EE,RR,BIY(J),ICLK,BALPHA)
CALL THTASL (EE,RR,PI,BALPHA,BTHETA)
```

```
JJIX(J)=T*SIGN
JJIIY(J)=(ABS(KKIY(J))-CHECK/3.0)*SIGN
JIX(J)=R*SIGN
JIIY(J)=JJIIY(J)
CIX(J)=XMAX*SIGN
CIY(J)=JJIIY(J)
```

```
CALL YALPSL (EE,RR,CIY(J),ICLK,CALPHA)
CALL THTASL (EE,RR,PI,CALPHA,CTHETA)
```

```
C *****
C *
C *          OVERLAP CONDITION FIVE
C *
C *    80% OF THE RIB LIES IN THE CYLINDRICAL REGION
C *    (QUADRANTS ONE AND THREE)
C *
C *****
```

```
ELSE IF(VRTICL.GT.4.0*CHECK/5.0) THEN
  WRITE(NO,*) 'PART 5'
  IF(PHIONE.LT.PI) THEN
    PHIB=PHIONE/3.0
    PHIC=2.0*PHIONE/3.0
  ELSE IF(PHIONE.GT.PI) THEN
    PHIC=PHIONE-(PHITWO-PHIONE)/4.0
    PHIB=PHIONE-2.0*(PHITWO-PHIONE)/5.0
  ENDIF
```

```
IIIX(J)=XXX(T,PHIB)
IIIIY(J)=YYY(T,PHIB)+TRANS
IIX(J)=XXX(R,PHIB)
IIY(J)=YYY(R,PHIB)+TRANS
```

```
CALL PALPSL (EE,RR,PI,PHIB,TRANS,BALPHA)
CALL THTASL (EE,RR,PI,BALPHA,BTHETA)
```

GEN10890  
GEN10900  
GEN10910  
GEN10920  
GEN10930  
GEN10940  
GEN10950  
GEN10960  
GEN10970  
GEN10980  
GEN10990  
GEN11000  
GEN11010  
GEN11020  
GEN11030  
GEN11040  
GEN11050  
GEN11060  
GEN11070  
GEN11080  
GEN11090  
GEN11100  
GEN11110  
GEN11120  
GEN11130  
GEN11140  
GEN11150  
GEN11160  
GEN11170  
GEN11180  
GEN11190  
GEN11200  
GEN11210  
GEN11220  
GEN11230  
GEN11240  
GEN11250  
GEN11260  
GEN11270  
GEN11280  
GEN11290  
GEN11300  
GEN11310  
GEN11320  
GEN11330  
GEN11340  
GEN11350  
GEN11360  
GEN11370  
GEN11380  
GEN11390  
GEN11400  
GEN11410  
GEN11420  
GEN11430  
GEN11440  
GEN11450  
GEN11460  
GEN11470  
GEN11480  
GEN11490  
GEN11500  
GEN11510  
GEN11520

```

      BIX(J)=FNX(BALPHA)
      BIY(J)=FNY(BALPHA)

      JJIX(J)=XXX(T,PHIC)
      JJIY(J)=YYY(T,PHIC)+TRANS
      JIX(J)=XXX(R,PHIC)
      JIY(J)=YYY(R,PHIC)+TRANS

      CALL PALPSL (EE,RR,PI,PHIC,TRANS,CALPHA)
      CALL THTASL (EE,RR,PI,CALPHA,CTHETA)

      CIX(J)=FNX(CALPHA)
      CIY(J)=FNY(CALPHA)

      *****
      *
      *          OVERLAP CONDITION SIX
      *
      *          BETWEEN 20% AND 80% OF THE RIB LIES IN THE
      *          CYLINDRICAL REGION
      *          (QUADRANTS ONE AND THREE)
      *
      *****
      ELSE IF ((VRTICL.GT.CHECK/5.).AND.(VRTICL.LT.4.*CHECK/5.)) THEN
        WRITE(NO,*) 'PART 6'
        IIIX(J)=T*SIGN
        IIIY(J)=((ABS(TRANS)-ABS(NNIY(J-1)))/2.0+ABS(NNIY(J-1)))*SIGN
        IIX(J)=R*SIGN
        IIY(J)=IIY(J)
        BIX(J)=XMAX*SIGN
        BIY(J)=IIY(J)

        CALL YALPSL (EE,RR,BIY(J),ICLK,BALPHA)
        CALL THTASL (EE,RR,PI,BALPHA,BTHETA)

        IF (PHIONE.LT.PI) THEN
          PHIC=PHIONE/2.0

        ELSE IF (PHIONE.GT.PI) THEN

          PHIC=PI+(PHIONE-PI)/2.0
        ENDIF

        JJIX(J)=XXX(T,PHIC)
        JJIY(J)=YYY(T,PHIC)+TRANS
        JIX(J)=XXX(R,PHIC)
        JIY(J)=YYY(R,PHIC)+TRANS

        CALL PALPSL (EE,RR,PI,PHIC,TRANS,CALPHA)
        CALL THTASL (EE,RR,PI,CALPHA,CTHETA)

        CIX(J)=FNX(CALPHA)
        CIY(J)=FNY(CALPHA)

      ENDIF
    ENDIF

    SIGN=ABS(SIGN)
    TRANS=ABS(TRANS)
    RECES=ABS(RECES)

    GO TO 110
```

```

GEN11530
GEN11540
GEN11550
GEN11560
GEN11570
GEN11580
GEN11590
GEN11600
GEN11610
GEN11620
GEN11630
GEN11640
GEN11650
GEN11660
GEN11670
GEN11680
GEN11690
GEN11700
GEN11710
GEN11720
GEN11730
GEN11740
GEN11750
GEN11760
GEN11770
GEN11780
GEN11790
GEN11800
GEN11810
GEN11820
GEN11830
GEN11840
GEN11850
GEN11860
GEN11870
GEN11880
GEN11890
GEN11900
GEN11910
GEN11920
GEN11930
GEN11940
GEN11950
GEN11960
GEN11970
GEN11980
GEN11990
GEN12000
GEN12010
GEN12020
GEN12030
GEN12040
GEN12050
GEN12060
GEN12070
GEN12080
GEN12090
GEN12100
GEN12110
GEN12120
GEN12130
GEN12140
GEN12150
GEN12160
```

```
105      IF (PHCHCK.EQ.1) THEN
          ICTA=ICTA+1
          IF (ICTA.EQ.1) ICHK=1
          IF (ICTA.GT.1) ICHK=0
      IF (SLSLVR.LT.1) THEN
          RECESX=-RECES
          RECESY=RECES
          SIGNX=-SIGN
          SIGNY=SIGN
      ELSE
          TRANS=-TRANS
          RECESX=RECES
          RECESY=-RECES
          SIGNX=SIGN
          SIGNY=-SIGN
      ENDIF

      IF (SLSLVR.GT.1) THEN
          PHITWI=PHITWO-2.0*PI
          PHITW2=2.0*PI
      ELSE
          PHITWI=PHITWO-PI
          PHITW2=PI
      ENDIF

      VRTICL=ABS (T*TAN (PHITWI))

      NNIX(J)=T*SIGNX
      NNIY(J)=(ABS (TRANS) -VRTICL) *SIGNY
      NIX(J)=R*SIGNX
      NIY(J)=NNIY(J)
      GIX(J)=XMAX*SIGNX
      GIY(J)=NNIY(J)

      IF (RIBTYP(J).EQ.2) THEN
          GRIX(J)=GIX(J)+RECESX
          GRIY(J)=GIY(J)+RECESY
      ENDIF

      CALL YALPSL (EE,RR,NIY(J),ICLK,GALPHA)
      CALL THTASL (EE,RR,PI,GALPHA,GTHETA)

      KKIX(J)=XXX(T,PHIONE)
      KKIY(J)=YYY(T,PHIONE)+TRANS
      KIX(J)=XXX(R,PHIONE)
      KIY(J)=YYY(R,PHIONE)+TRANS

      CALL PALPSL (EE,RR,PI,PHIONE,TRANS,DALPHA)
      CALL THTASL (EE,RR,PI,DALPHA,DTHETA)

      DIX(J)=FNX(DALPHA)
      DIY(J)=FNY(DALPHA)

      IF (RIBTYP(J).EQ.2) THEN
          DRIX(J)=DIX(J)+RECES*cos (PHID)
```

```
GEN12170
GEN12180
GEN12190
GEN12200
GEN12210
GEN12220
GEN12230
GEN12240
GEN12250
GEN12260
GEN12270
GEN12280
GEN12290
GEN12300
GEN12310
GEN12320
GEN12330
GEN12340
GEN12350
GEN12360
GEN12370
GEN12380
GEN12390
GEN12400
GEN12410
GEN12420
GEN12430
GEN12440
GEN12450
GEN12460
GEN12470
GEN12480
GEN12490
GEN12500
GEN12510
GEN12520
GEN12530
GEN12540
GEN12550
GEN12560
GEN12570
GEN12580
GEN12590
GEN12600
GEN12610
GEN12620
GEN12630
GEN12640
GEN12650
GEN12660
GEN12670
GEN12680
GEN12690
GEN12700
GEN12710
GEN12720
GEN12730
GEN12740
GEN12750
GEN12760
GEN12770
GEN12780
GEN12790
GEN12800
```

```

      DRIY(J)=DIY(J)+RECES*SIN(PHID)
      ENDIF
      CHECK=ABS(ABS(KKIY(J))-ABS(NNIY(J)))
C *****
C *
C *          OVERLAP CONDITION SEVEN
C *
C *      20% OF THE RIB LIES IN THE CYLINDRICAL REGION
C *          (QUADRANTS TWO AND FOUR)
C *
C *****
      IF(VRTICL.LT.CHECK/5.0) THEN
        WRITE(NO,*) 'PART 7'
        PHIE=(PHITW2-PHIONE)/3.0+PHIONE
        PHIF=2.0*(PHITW2-PHIONE)/3.0+PHIONE
        LLIX(J)=XXX(T,PHIE)
        LLIY(J)=YYY(T,PHIE)+TRANS
        LIX(J)=XXX(R,PHIE)
        LIY(J)=YYY(R,PHIE)+TRANS
        CALL PALPSL (EE,RR,PI,PHIE,TRANS,EALPHA)
        CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
        EIX(J)=FNX(EALPHA)
        EIY(J)=FNY(EALPHA)
        IF(RIBTYP(J).EQ.2) THEN
          ERIX(J)=EIX(J)+RECES*COS(PHIE)
          ERIY(J)=EIY(J)+RECES*SIN(PHIE)
        ENDIF
        MMIX(J)=XXX(T,PHIF)
        MMIY(J)=YYY(T,PHIF)+TRANS
        MIX(J)=XXX(R,PHIF)
        MIY(J)=YYY(R,PHIF)+TRANS
        CALL PALPSL (EE,RR,PI,PHIF,TRANS,FALPHA)
        CALL THTASL (EE,RR,PI,FALPHA,FTHETA)
        FIX(J)=FNX(FALPHA)
        FIY(J)=FNY(FALPHA)
        IF(RIBTYP(J).EQ.2) THEN
          ERIX(J)=EIX(J)+RECES*COS(PHIE)
          ERIY(J)=EIY(J)+RECES*SIN(PHIE)
        ENDIF
C *****
C *
C *          OVERLAP CONDITION EIGHT
C *
C *      80% OF THE RIB LIES IN THE CYLINDRICAL REGION
C *          (QUADRANTS TWO AND FOUR)
C *
C *****
      GEN12810
      GEN12820
      GEN12830
      GEN12840
      GEN12850
      GEN12860
      GEN12870
      GEN12880
      GEN12890
      GEN12900
      GEN12910
      GEN12920
      GEN12930
      GEN12940
      GEN12950
      GEN12960
      GEN12970
      GEN12980
      GEN12990
      GEN13000
      GEN13010
      GEN13020
      GEN13030
      GEN13040
      GEN13050
      GEN13060
      GEN13070
      GEN13080
      GEN13090
      GEN13100
      GEN13110
      GEN13120
      GEN13130
      GEN13140
      GEN13150
      GEN13160
      GEN13170
      GEN13180
      GEN13190
      GEN13200
      GEN13210
      GEN13220
      GEN13230
      GEN13240
      GEN13250
      GEN13260
      GEN13270
      GEN13280
      GEN13290
      GEN13300
      GEN13310
      GEN13320
      GEN13330
      GEN13340
      GEN13350
      GEN13360
      GEN13370
      GEN13380
      GEN13390
      GEN13400
      GEN13410
      GEN13420
      GEN13430
      GEN13440
```

```
ELSE IF (VRTICL.GT.4.0*CHECK/5.0) THEN
```

```
  WRITE(NO,*) 'PART 8'
```

```
    LLIX(J)=T*SIGNX
```

```
    LLIY(J)=(ABS(NNIY(J))+2.0*(VRTICL)/3.0)*SIGNY
```

```
    LIX(J)=R*SIGNX
```

```
    LIY(J)=LLIY(J)
```

```
    EIX(J)=XMAX*SIGNX
```

```
    EIIY(J)=LLIY(J)
```

```
  IF (RIBTYP(J).EQ.2) THEN
```

```
    ERIX(J)=EIX(J)+RECESX
```

```
    ERIY(J)=EIIY(J)+RECESY
```

```
  ENDIF
```

```
  CALL YALPSL (EE,RR,LIY(J),ICLK,EALPHA)
```

```
  CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
```

```
    MMIX(J)=T*SIGNX
```

```
    MMIY(J)=(ABS(NNIY(J))+(VRTICL)/3.0)*SIGNY
```

```
    MIX(J)=R*SIGNX
```

```
    MIY(J)=MMIY(J)
```

```
    FIX(J)=XMAX*SIGNX
```

```
    FIY(J)=MMIY(J)
```

```
  IF (RIBTYP(J).EQ.2) THEN
```

```
    FRIX(J)=FIX(J)+RECESX
```

```
    FRIY(J)=FIY(J)+RECESY
```

```
  ENDIF
```

```
  CALL YALPSL (EE,RR,MIY(J),ICLK,FALPHA)
```

```
  CALL THTASL (EE,RR,PI,FALPHA,FTHETA)
```

```
*****
C  *
C  *          OVERLAP CONDITION NINE
C  *
C  *          BETWEEN 20% AND 80% OF THE RIB LIES IN THE
C  *          CYLINDRICAL REGION
C  *          (QUADRANTS TWO AND FOUR)
C  *
C  *****
```

```
ELSE IF ((VRTICL.GT.CHECK/5.0).AND.(VRTICL.LT.4.0*CHECK/5.0)) THEN
```

```
  WRITE(NO,*) 'PART 9'
```

```
  IF (SLSLVR.GT.1) THEN
```

```
    PHIE=(2.0*PI-PHIONE)/2.0+PHIONE
```

```
  ELSE
```

```
    PHIE=(PI-PHIONE)/2.0+PHIONE
```

```
  ENDIF
```

```
    LLIX(J)=XXX(T,PHIE)
```

```
    LLIY(J)=YYY(T,PHIE)+TRANS
```

```
    LIX(J)=XXX(R,PHIE)
```

```
    LIY(J)=YYY(R,PHIE)+TRANS
```

```
  CALL PALPSL (EE,RR,PI,PHIE,TRANS,FALPHA)
```

```
  CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
```

GEN13450  
 GEN13460  
 GEN13470  
 GEN13480  
 GEN13490  
 GEN13500  
 GEN13510  
 GEN13520  
 GEN13530  
 GEN13540  
 GEN13550  
 GEN13560  
 GEN13570  
 GEN13580  
 GEN13590  
 GEN13600  
 GEN13610  
 GEN13620  
 GEN13630  
 GEN13640  
 GEN13650  
 GEN13660  
 GEN13670  
 GEN13680  
 GEN13690  
 GEN13700  
 GEN13710  
 GEN13720  
 GEN13730  
 GEN13740  
 GEN13750  
 GEN13760  
 GEN13770  
 GEN13780  
 GEN13790  
 GEN13800  
 GEN13810  
 GEN13820  
 GEN13830  
 GEN13840  
 GEN13850  
 GEN13860  
 GEN13870  
 GEN13880  
 GEN13890  
 GEN13900  
 GEN13910  
 GEN13920  
 GEN13930  
 GEN13940  
 GEN13950  
 GEN13960  
 GEN13970  
 GEN13980  
 GEN13990  
 GEN14000  
 GEN14010  
 GEN14020  
 GEN14030  
 GEN14040  
 GEN14050  
 GEN14060  
 GEN14070  
 GEN14080



```

EIX(J)=FNX(EALPHA)
EIY(J)=FNY(EALPHA)

IF(RIBTYP(J).EQ.2) THEN

    ERIX(J)=EIX(J)+RECES*COS(PHIE)
    ERIY(J)=EIY(J)+RECES*SIN(PHIE)

ENDIF

MMIX(J)=T*SIGNX
MMIY(J)=(ABS(NNIY(J))+VRTICL/2.0)*SIGNY
MIX(J)=R*SIGNX
MIY(J)=MMIY(J)
FIX(J)=XMAX*SIGNX
FIY(J)=MMIY(J)

IF(RIBTYP(J).EQ.2) THEN

    FRIX(J)=FIX(J)+RECESX
    FRIY(J)=FIY(J)+RECESY

ENDIF

CALL YALPSL (EE,RR,FIY(J),ICLK,FALPHA)
CALL THTASL (EE,RR,PI,FALPHA,FTHETA)

ENDIF

*****
*
*   THIS SECTION CALCULATES THE CHANNEL COORDINATES OF
*   OVERLAP CONDITIONS SEVEN THROUGH NINE
*
*****

PHIB=PHIT+(PHIONE-PHIT)/3.0
PHIC=PHIT+2.0*(PHIONE-PHIT)/3.0

IIIX(J)=XXX(T,PHIB)
IIY(J)=YYY(T,PHIB)+TRANS
IIX(J)=XXX(R,PHIB)
IIY(J)=YYY(R,PHIB)+TRANS

CALL PALPSL (EE,RR,PI,PHIB,TRANS,BALPHA)
CALL THTASL (EE,RR,PI,BALPHA,BTHETA)

BIX(J)=FNX(BALPHA)
BIY(J)=FNY(BALPHA)

JJIX(J)=XXX(T,PHIC)
JJIY(J)=YYY(T,PHIC)+TRANS
JIX(J)=XXX(R,PHIC)
JIY(J)=YYY(R,PHIC)+TRANS

CALL PALPSL (EE,RR,PI,PHIC,TRANS,CALPHA)
CALL THTASL (EE,RR,PI,CALPHA,CTHETA)

CIX(J)=FNX(CALPHA)
CIY(J)=FNY(CALPHA)

HIX(J)=NIX(J-1)
HIY(J)=NIY(J-1)

```

```

GEN14090
GEN14100
GEN14110
GEN14120
GEN14130
GEN14140
GEN14150
GEN14160
GEN14170
GEN14180
GEN14190
GEN14200
GEN14210
GEN14220
GEN14230
GEN14240
GEN14250
GEN14260
GEN14270
GEN14280
GEN14290
GEN14300
GEN14310
GEN14320
GEN14330
GEN14340
GEN14350
GEN14360
GEN14370
GEN14380
GEN14390
GEN14400
GEN14410
GEN14420
GEN14430
GEN14440
GEN14450
GEN14460
GEN14470
GEN14480
GEN14490
GEN14500
GEN14510
GEN14520
GEN14530
GEN14540
GEN14550
GEN14560
GEN14570
GEN14580
GEN14590
GEN14600
GEN14610
GEN14620
GEN14630
GEN14640
GEN14650
GEN14660
GEN14670
GEN14680
GEN14690
GEN14700
GEN14710
GEN14720

```

```
C *****
C *
C *          END CHANNEL COORDINATE CALCULATION
C *
C *****
      ELSE IF (PHCHCK.EQ.2) THEN
        IF (SLSLVR.GT.1) THEN
          VRTICL=T*TAN(PHIONE-2.0*PI)
        ELSE
          VRTICL=T*TAN(PHIONE-PI)
        ENDIF

        KKIX(J)=T*SIGNX
        KKIY(J)=(TRANS-VRTICL)*SIGNY
        KIX(J)=R*SIGNX
        KIY(J)=KKIY(J)
        DIX(J)=XMAX*SIGNX
        DIY(J)=KKIY(J)

        IF (RIBTYP(J).EQ.2) THEN
          DRIX(J)=DIX(J)+RECESX
          DRIY(J)=DIY(J)+RECESY
        ENDIF

        CALL YALPSL (EE,RR,DIY(J),ICLK,DALPHA)
        CALL THTASL (EE,RR,PI,DALPHA,DTHETA)

        NNIX(J)=T*SIGNX
        NNIY(J)=(TRANS-VRTICL)*SIGNY
        NIX(J)=R*SIGNX
        NIY(J)=NNIY(J)
        GIX(J)=XMAX*SIGNX
        GIY(J)=NNIY(J)

        IF (RIBTYP(J).EQ.2) THEN
          GRIX(J)=GIX(J)+RECESX
          GRIY(J)=GIY(J)+RECESY
        ENDIF

        CALL YALPSL (EE,RR,GIY(J),ICLK,GALPHA)
        CALL THTASL (EE,RR,PI,GALPHA,GTHETA)

        TRIB=KKIY(J)-NNIY(J)

        LLIY(J)=T*SIGNX
        LLIY(J)=(NNIX(J)+2.0*TRIB/3.0)*SIGNY
        LIX(J)=R*SIGNX
        LIY(J)=LLIY(J)
        EIX(J)=XMAX*SIGNX
        EIIY(J)=LLIY(J)

        IF (RIBTYP(J).EQ.2) THEN
          ERIX(J)=EIX(J)+RECESX
          ERIY(J)=EIIY(J)+RECESY
        ENDIF
```

GEN14730  
GEN14740  
GEN14750  
GEN14760  
GEN14770  
GEN14780  
GEN14790  
GEN14800  
GEN14810  
GEN14820  
GEN14830  
GEN14840  
GEN14850  
GEN14860  
GEN14870  
GEN14880  
GEN14890  
GEN14900  
GEN14910  
GEN14920  
GEN14930  
GEN14940  
GEN14950  
GEN14960  
GEN14970  
GEN14980  
GEN14990  
GEN15000  
GEN15010  
GEN15020  
GEN15030  
GEN15040  
GEN15050  
GEN15060  
GEN15070  
GEN15080  
GEN15090  
GEN15100  
GEN15110  
GEN15120  
GEN15130  
GEN15140  
GEN15150  
GEN15160  
GEN15170  
GEN15180  
GEN15190  
GEN15200  
GEN15210  
GEN15220  
GEN15230  
GEN15240  
GEN15250  
GEN15260  
GEN15270  
GEN15280  
GEN15290  
GEN15300  
GEN15310  
GEN15320  
GEN15330  
GEN15340  
GEN15350  
GEN15360

```
CALL YALPSL (EE,RR,EIY(J),ICLK,EALPHA)
CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
```

```
MMIX(J)=T*SIGNX
MMIY(J)=(NNIY(J)+TRIB/3.0)*SIGNY
MIX(J)=R*SIGNX
MIY(J)=MMIY(J)
FIX(J)=XMAX*SIGNX
FIY(J)=MMIY(J)
```

```
IF(RIBTYP(J).EQ.2) THEN
```

```
FRIX(J)=FIX(J)+RECESX
FRIY(J)=FIY(J)+RECESY
```

```
ENDIF
```

```
CALL YALPSL (EE,RR,FIY(J),ICLK,FALPHA)
CALL THTASL (EE,RR,PI,FALPHA,FTHETA)
```

```
HIX(J)=NIX(J-1)
HIY(J)=NIY(J-1)
```

```
CHECK=TRANS-KKIY(J)
```

```
C *****
C *
C *          OVERLAP CONDITION TEN
C *
C *  20% OF THE CHANNEL LIES IN THE CYLINDRICAL REGION
C *          (QUADRANTS TWO AND FOUR)
C *
C *****
```

```
IF(CHECK.LT.TRIB/5.0) THEN
```

```
WRITE(NO,*) 'PART 10'
```

```
ANGLE=NNIY(J-1)/NNIX(J-1)
PHIH=ATAN(ANGLE)
```

```
PHIB=PHIH+(PHIONE-PHIH)/3.0
PHIC=PHIH+2.0*(PHIONE-PHIH)/3.0
```

```
IIIX(J)=XXX(T,PHIB)
IIY(J)=YYY(T,PHIB)+TRANS
IIX(J)=XXX(R,PHIB)
IIY(J)=YYY(R,PHIB)+TRANS
```

```
CALL PALPSL (EE,RR,PI,PHIB,TRANS,BALPHA)
CALL THTASL (EE,RR,PI,BALPHA,BTHETA)
```

```
BIX(J)=FNX(BALPHA)
BIY(J)=FNY(BALPHA)
```

```
JJIX(J)=XXX(T,PHIC)
JJY(J)=YYY(T,PHIC)+TRANS
JIX(J)=XXX(R,PHIC)
JIY(J)=YYY(R,PHIC)+TRANS
```

```
CALL PALPSL (EE,RR,PI,PHIC,TRANS,CALPHA)
CALL THTASL (EE,RR,PI,CALPHA,CTHETA)
```

```
CIX(J)=FNX(CALPHA)
CIY(J)=FNY(CALPHA)
```

GEN15370  
GEN15380  
GEN15390  
GEN15400  
GEN15410  
GEN15420  
GEN15430  
GEN15440  
GEN15450  
GEN15460  
GEN15470  
GEN15480  
GEN15490  
GEN15500  
GEN15510  
GEN15520  
GEN15530  
GEN15540  
GEN15550  
GEN15560  
GEN15570  
GEN15580  
GEN15590  
GEN15600  
GEN15610  
GEN15620  
GEN15630  
GEN15640  
GEN15650  
GEN15660  
GEN15670  
GEN15680  
GEN15690  
GEN15700  
GEN15710  
GEN15720  
GEN15730  
GEN15740  
GEN15750  
GEN15760  
GEN15770  
GEN15780  
GEN15790  
GEN15800  
GEN15810  
GEN15820  
GEN15830  
GEN15840  
GEN15850  
GEN15860  
GEN15870  
GEN15880  
GEN15890  
GEN15900  
GEN15910  
GEN15920  
GEN15930  
GEN15940  
GEN15950  
GEN15960  
GEN15970  
GEN15980  
GEN15990  
GEN16000

```
C *****
C *
C *          OVERLAP CONDITION ELEVEN
C *
C *      80% OF THE CHANNEL LIES IN THE CYLINDRICAL REGION
C *      (QUADRANTS TWO AND FOUR)
C *
C *****

      ELSE IF (CHECK.GT.4.0*TRIB/5.0) THEN
        WRITE(NO,*) 'PART 11'
        IIIX(J)=T*SIGNX
        IIIY(J)=(KKIY(J)+2.0*TRIB/3.0)*SIGNY
        IIX(J)=R*SIGNX
        IIY(J)=IIY(J)
        BIX(J)=XMAX*SIGNX
        BIY(J)=IIY(J)

        CALL YALPSL (EE,RR,BIY(J),ICLK,BALPHA)
        CALL THTASL (EE,RR,PI,BALPHA,BTHETA)

        JJIX(J)=T*SIGNX
        JJIY(J)=(KKIY(J)+TRIB/3.0)*SIGNY
        JIX(J)=R*SIGNX
        JIY(J)=JJIY(J)
        CIX(J)=XMAX*SIGNX
        CIY(J)=JJIY(J)

        CALL YALPSL (EE,RR,CIY(J),ICLK,CALPHA)
        CALL THTASL (EE,RR,PI,CALPHA,CTHETA)

C *****
C *
C *          OVERLAP CONDITION TWELVE
C *
C *      BETWEEN 20% AND 80% OF THE CHANNEL LIES IN THE
C *      CYLINDRICAL REGION
C *      (QUADRANTS TWO AND FOUR)
C *
C *****

      ELSE IF ((CHECK.GT.TRIB/5.0).AND.(CHECK.LT.4.0*TRIB/5.0)) THEN
        WRITE(NO,*) 'PART 12'
        ANGLE=NNIY(J-1)/NNIY(J-1)
        PHIH=ATAN(ANGLE)

        PHIB=PHIH+(PHIH+PHIONE)/2.0

        IIIX(J)=XXX(T,PHIB)
        IIIY(J)=YYY(T,PHIB)+TRANS
        IIX(J)=XXX(R,PHIB)
        IIY(J)=YYY(R,PHIB)+TRANS

        CALL PALPSL (EE,RR,PI,PHIB,TRANS,BALPHA)
        CALL THTASL (EE,RR,PI,BALPHA,BTHETA)

        BIX(J)=FNX(BALPHA)
        BIY(J)=FNY(BALPHA)

        JJIX(J)=T*SIGNX
        JJIY(J)=(KKIY(J)+CHECK/2.0)*SIGNY
        JIX(J)=R*SIGNX
        JIY(J)=JJIY(J)
```

GEN16010  
GEN16020  
GEN16030  
GEN16040  
GEN16050  
GEN16060  
GEN16070  
GEN16080  
GEN16090  
GEN16100  
GEN16110  
GEN16120  
GEN16130  
GEN16140  
GEN16150  
GEN16160  
GEN16170  
GEN16180  
GEN16190  
GEN16200  
GEN16210  
GEN16220  
GEN16230  
GEN16240  
GEN16250  
GEN16260  
GEN16270  
GEN16280  
GEN16290  
GEN16300  
GEN16310  
GEN16320  
GEN16330  
GEN16340  
GEN16350  
GEN16360  
GEN16370  
GEN16380  
GEN16390  
GEN16400  
GEN16410  
GEN16420  
GEN16430  
GEN16440  
GEN16450  
GEN16460  
GEN16470  
GEN16480  
GEN16490  
GEN16500  
GEN16510  
GEN16520  
GEN16530  
GEN16540  
GEN16550  
GEN16560  
GEN16570  
GEN16580  
GEN16590  
GEN16600  
GEN16610  
GEN16620  
GEN16630  
GEN16640

```
CIX(J)=XMAX*SIGNX
CIY(J)=JJIY(J)
```

```
CALL YALPSL (EE,RR,CIY(J),ICLK,CALPHA)
CALL THTASL (EE,RR,PI,CALPHA,CTHETA)
```

```
ENDIF
```

```
ENDIF
```

```
C *****
C *
C * THIS SECTION CALCUALTES THE INNER SHELL COORDINATES
C * FOR THE OVERLAP CONDITIONS
C *
C *****
```

```
110 BBBIX(J)=FINERX(BALPHA,BTHETA,CC)
    BBBIY(J)=FINERY(BALPHA,BTHETA,CC)
    BBIX(J)=FINERX(BALPHA,BTHETA,BB)
    BBIIY(J)=FINERY(BALPHA,BTHETA,BB)
```

```
    CCCIX(J)=FINERX(CALPHA,CTHETA,CC)
    CCCIY(J)=FINERY(CALPHA,CTHETA,CC)
    CCIX(J)=FINERX(CALPHA,CTHETA,BB)
    CCIY(J)=FINERY(CALPHA,CTHETA,BB)
```

```
    DDDIX(J)=FINERX(DALPHA,DTHETA,CC)
    DDDIY(J)=FINERY(DALPHA,DTHETA,CC)
    DDIX(J)=FINERX(DALPHA,DTHETA,BB)
    DDIY(J)=FINERY(DALPHA,DTHETA,BB)
```

```
    EEEIX(J)=FINERX(EALPHA,ETHETA,CC)
    EEEIY(J)=FINERY(EALPHA,ETHETA,CC)
    EEIX(J)=FINERX(EALPHA,ETHETA,BB)
    EEIY(J)=FINERY(EALPHA,ETHETA,BB)
```

```
    FFFIX(J)=FINERX(FALPHA,FTHETA,CC)
    FFFIY(J)=FINERY(FALPHA,FTHETA,CC)
    FFIIX(J)=FINERX(FALPHA,FTHETA,BB)
    FFIY(J)=FINERY(FALPHA,FTHETA,BB)
```

```
    GGGIX(J)=FINERX(GALPHA,GTHETA,CC)
    GGGIY(J)=FINERY(GALPHA,GTHETA,CC)
    GGIX(J)=FINERX(GALPHA,GTHETA,BB)
    GGIY(J)=FINERY(GALPHA,GTHETA,BB)
```

```
TRANS=ABS(TRANS)
GO TO 125
```

```
C *****
C *
C * END OF OVERLAPPING RIB/CAHNNEL POINT CALCULATION
C *
C *****
```

```
C *****
C *
C * THIS SECTION CALCUALTES THE POINT COORDINATES OF
C * THE RIB/CHANNEL SEGMENTS THAT LIE IN THE
C * RECTANGULAR REGION ON THE RIGHT SIDE OF THE HOUSING
C *
C *****
```

```
115 SLSLVR=SLSLVR+2
```

GEN16650  
GEN16660  
GEN16670  
GEN16680  
GEN16690  
GEN16700  
GEN16710  
GEN16720  
GEN16730  
GEN16740  
GEN16750  
GEN16760  
GEN16770  
GEN16780  
GEN16790  
GEN16800  
GEN16810  
GEN16820  
GEN16830  
GEN16840  
GEN16850  
GEN16860  
GEN16870  
GEN16880  
GEN16890  
GEN16900  
GEN16910  
GEN16920  
GEN16930  
GEN16940  
GEN16950  
GEN16960  
GEN16970  
GEN16980  
GEN16990  
GEN17000  
GEN17010  
GEN17020  
GEN17030  
GEN17040  
GEN17050  
GEN17060  
GEN17070  
GEN17080  
GEN17090  
GEN17100  
GEN17110  
GEN17120  
GEN17130  
GEN17140  
GEN17150  
GEN17160  
GEN17170  
GEN17180  
GEN17190  
GEN17200  
GEN17210  
GEN17220  
GEN17230  
GEN17240  
GEN17250  
GEN17260  
GEN17270  
GEN17280

```
IF (GIX(J-1).GT.0.0) THEN
WRITE(NO,*) 'RIGHT SIDE'
LINK=0
```

```
TRIB=ABS(YTWO-YONE)
```

```
AIY(J)=GIY(J-1)
DIY(J)=YONE
DDIY(J)=DIY(J)
DDDIY(J)=DIY(J)
BIY(J)=AIY(J)+ABS(DIY(J)-AIY(J))/3.0
BBIY(J)=BIY(J)
BBBIY(J)=BIY(J)
CIY(J)=AIY(J)+2.0*ABS(DIY(J)-AIY(J))/3.0
CCIY(J)=CIY(J)
CCCIY(J)=CIY(J)
```

```
GIY(J)=YTWO
GGIY(J)=GIY(J)
GGGIY(J)=GIY(J)
EIIY(J)=DIY(J)+ABS(GIY(J)-DIY(J))/3.0
EEIY(J)=EIIY(J)
EEEIY(J)=EIIY(J)
FIY(J)=DIY(J)+2.0*ABS(GIY(J)-DIY(J))/3.0
FFIY(J)=FIY(J)
FFFIY(J)=FIY(J)
```

```
C *****
C *
C * THIS SECTION CALCUALTES THE POINT COORDINATES OF *
C * THE RIB/CHANNEL SEGMENTS IF IT LIES IN THE *
C * RECTANGULAR REGION ON THE LEFT SIDE OF THE HOUSING *
C *****
```

```
ELSE IF (GIX(J-1).LT.0.0) THEN
```

```
WRITE(NO,*) 'LEFT SIDE'
```

```
LINK=1
```

```
AIY(J)=GIY(J-1)
DIY(J)=YONE
DDIY(J)=DIY(J)
DDDIY(J)=DIY(J)
BIY(J)=AIY(J)-ABS(DIY(J)-AIY(J))/3.0
BBIY(J)=BIY(J)
BBBIY(J)=BIY(J)
CIY(J)=AIY(J)-2.0*ABS(DIY(J)-AIY(J))/3.0
CCIY(J)=CIY(J)
CCCIY(J)=CIY(J)
```

```
GIY(J)=YTWO
GGIY(J)=GIY(J)
GGGIY(J)=GIY(J)
EIIY(J)=DIY(J)-ABS(GIY(J)-DIY(J))/3.0
EEIY(J)=EIIY(J)
EEEIY(J)=EIIY(J)
FIY(J)=DIY(J)-2.0*ABS(GIY(J)-DIY(J))/3.0
FFIY(J)=FIY(J)
FFFIY(J)=FIY(J)
```

```
GEN17290
GEN17300
GEN17310
GEN17320
GEN17330
GEN17340
GEN17350
GEN17360
GEN17370
GEN17380
GEN17390
GEN17400
GEN17410
GEN17420
GEN17430
GEN17440
GEN17450
GEN17460
GEN17470
GEN17480
GEN17490
GEN17500
GEN17510
GEN17520
GEN17530
GEN17540
GEN17550
GEN17560
GEN17570
GEN17580
GEN17590
GEN17600
GEN17610
GEN17620
GEN17630
GEN17640
GEN17650
GEN17660
GEN17670
GEN17680
GEN17690
GEN17700
GEN17710
GEN17720
GEN17730
GEN17740
GEN17750
GEN17760
GEN17770
GEN17780
GEN17790
GEN17800
GEN17810
GEN17820
GEN17830
GEN17840
GEN17850
GEN17860
GEN17870
GEN17880
GEN17890
GEN17900
GEN17910
GEN17920
```

```
ENDIF
      IF (LINK.EQ.1) SIGN=-SIGN

      HIY(J)=NIY(J-1)
      HHIY(J)=NNIY(J-1)
      Iiy(J)=BIY(J)
      IIIY(J)=BIY(J)
      JIY(J)=CIY(J)
      JJIY(J)=CIY(J)
      KIY(J)=DIY(J)
      KKIY(J)=DIY(J)
      LIY(J)=EIY(J)
      LLIY(J)=EIY(J)
      MIY(J)=FIY(J)
      MMIY(J)=FIY(J)
      NIY(J)=GIY(J)
      NNIY(J)=GIY(J)

      BIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,BIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      BBIX(J)=FINERX (ALPHA, THETA, BB)
      BBBIX(J)=FINERX (ALPHA, THETA, CC)

      CIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,CIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      CCIX(J)=FINERX (ALPHA, THETA, BB)
      CCCIX(J)=FINERX (ALPHA, THETA, CC)

      DIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,DIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      DDIX(J)=FINERX (ALPHA, THETA, BB)
      DDDIX(J)=FINERX (ALPHA, THETA, CC)

      EIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,EIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      EEIX(J)=FINERX (ALPHA, THETA, BB)
      EEEIX(J)=FINERX (ALPHA, THETA, CC)

      FIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,FIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      FFIX(J)=FINERX (ALPHA, THETA, BB)
      FFFIX(J)=FINERX (ALPHA, THETA, CC)

      GIX(J)=XMAX*SIGN
      CALL YALPSL (EE,RR,GIY(J), ICHK, ALPHA)
      CALL THTASL (EE,RR,PI, ALPHA, THETA)
      GGIX(J)=FINERX (ALPHA, THETA, BB)
      GGGIX(J)=FINERX (ALPHA, THETA, CC)

      HIX(J)=NIX(J-1)
      HHIX(J)=NNIX(J-1)
      IIX(J)=HIX(J)
      IIIX(J)=HHIX(J)
      JIX(J)=IIX(J)
      JJIX(J)=IIIX(J)
      KIX(J)=JIX(J)
      KKIX(J)=JJIX(J)
      LIX(J)=KIX(J)
      LLIX(J)=KKIX(J)
```

GEN17930  
GEN17940  
GEN17950  
GEN17960  
GEN17970  
GEN17980  
GEN17990  
GEN18000  
GEN18010  
GEN18020  
GEN18030  
GEN18040  
GEN18050  
GEN18060  
GEN18070  
GEN18080  
GEN18090  
GEN18100  
GEN18110  
GEN18120  
GEN18130  
GEN18140  
GEN18150  
GEN18160  
GEN18170  
GEN18180  
GEN18190  
GEN18200  
GEN18210  
GEN18220  
GEN18230  
GEN18240  
GEN18250  
GEN18260  
GEN18270  
GEN18280  
GEN18290  
GEN18300  
GEN18310  
GEN18320  
GEN18330  
GEN18340  
GEN18350  
GEN18360  
GEN18370  
GEN18380  
GEN18390  
GEN18400  
GEN18410  
GEN18420  
GEN18430  
GEN18440  
GEN18450  
GEN18460  
GEN18470  
GEN18480  
GEN18490  
GEN18500  
GEN18510  
GEN18520  
GEN18530  
GEN18540  
GEN18550  
GEN18560

```
MIX(J)=LIX(J)
MMIX(J)=LLIX(J)
NIX(J)=MIX(J)
NNIX(J)=MMIX(J)

IF(RIBTYP(J).EQ.2) THEN

  IF(DIX(J).LT.0.0) RECES=-RECES

  DRIX(J)=DIX(J)+RECES
  DRIY(J)=DIY(J)
  ERIX(J)=EIX(J)+RECES
  ERIY(J)=EIY(J)
  FRIX(J)=FIX(J)+RECES
  FRIY(J)=FIY(J)
  GRIX(J)=GIX(J)+RECES
  GRIY(J)=GIY(J)

ENDIF

SIGN=ABS(SIGN)
TRANS=ABS(TRANS)
GO TO 125

C *****
C *
C *          END OF RECTANGULAR REGION RIB/CHANNEL
C *          POINT CALULATION
C *
C *****
C *****
C *
C *          THIS SECTION CALCUALTES THE POINT COORDINATES OF
C *          THE RIB/CHANNEL SEGMENTS IF IT LIES IN THE
C *          CYLINDRICAL REGION
C *
C *****
120  PHIA=PHIG
    PHIA1=PHIA
    PHIB=PHIA+(PHIONE-PHIA)/3.0
    PHIB1=PHIB
    PHIC=PHIA+2.0*(PHIONE-PHIA)/3.0
    PHIC1=PHIC
    PHID=PHIONE
    PHID1=PHID
    PHIE=PHIONE+(PHITWO-PHIONE)/3.0
    PHIE1=PHIE
    PHIF=PHIONE+2.0*(PHITWO-PHIONE)/3.0
    PHIF1=PHIF
    PHIG=PHITWO
    PHIG1=PHIG
    SIGN1=1.0
    IF(PHIA.GT.PI) THEN
      PHIA1=2.0*PI-PHIA
      PHIB1=2.0*PI-PHIB
      PHIC1=2.0*PI-PHIC
      PHID1=2.0*PI-PHID
      PHIE1=2.0*PI-PHIE
      PHIF1=2.0*PI-PHIF
      PHIG1=2.0*PI-PHIG
      SIGN1=-1.0
    
```

```
GEN18570
GEN18580
GEN18590
GEN18600
GEN18610
GEN18620
GEN18630
GEN18640
GEN18650
GEN18660
GEN18670
GEN18680
GEN18690
GEN18700
GEN18710
GEN18720
GEN18730
GEN18740
GEN18750
GEN18760
GEN18770
GEN18780
GEN18790
GEN18800
GEN18810
GEN18820
GEN18830
GEN18840
GEN18850
GEN18860
GEN18870
GEN18880
GEN18890
GEN18900
GEN18910
GEN18920
GEN18930
GEN18940
GEN18950
GEN18960
GEN18970
GEN18980
GEN18990
GEN19000
GEN19010
GEN19020
GEN19030
GEN19040
GEN19050
GEN19060
GEN19070
GEN19080
GEN19090
GEN19100
GEN19110
GEN19120
GEN19130
GEN19140
GEN19150
GEN19160
GEN19170
GEN19180
GEN19190
GEN19200
```



```
ENDIF

CALL PALPSL (EE,RR,PI,PHIA1,TRANS,AALPHA)
CALL THTASL (EE,RR,PI,AALPHA,ATHETA)
CALL PALPSL (EE,RR,PI,PHIB1,TRANS,BALPHA)
CALL THTASL (EE,RR,PI,BALPHA,BTHETA)
CALL PALPSL (EE,RR,PI,PHIC1,TRANS,CALPHA)
CALL THTASL (EE,RR,PI,CALPHA,CTHETA)
CALL PALPSL (EE,RR,PI,PHID1,TRANS,DALPHA)
CALL THTASL (EE,RR,PI,DALPHA,DTHETA)
CALL PALPSL (EE,RR,PI,PHIE1,TRANS,EALPHA)
CALL THTASL (EE,RR,PI,EALPHA,ETHETA)
CALL PALPSL (EE,RR,PI,PHIF1,TRANS,FALPHA)
CALL THTASL (EE,RR,PI,FALPHA,FTHETA)
CALL PALPSL (EE,RR,PI,PHIG1,TRANS,GALPHA)
CALL THTASL (EE,RR,PI,GALPHA,GTHETA)

BIX(J)=FNX(BALPHA)
BIY(J)=FNY(BALPHA)*SIGN1
BBIX(J)=FINERX(BALPHA,BTHETA,BB)
BBIY(J)=FINERY(BALPHA,BTHETA,BB)*SIGN1
BBBIX(J)=FINERX(BALPHA,BTHETA,CC)
BBBIY(J)=FINERY(BALPHA,BTHETA,CC)*SIGN1

CIX(J)=FNX(CALPHA)
CIY(J)=FNY(CALPHA)*SIGN1
CCIX(J)=FINERX(CALPHA,CTHETA,BB)
CCIY(J)=FINERY(CALPHA,CTHETA,BB)*SIGN1
CCCIX(J)=FINERX(CALPHA,CTHETA,CC)
CCCIY(J)=FINERY(CALPHA,CTHETA,CC)*SIGN1

DIX(J)=FNX(DALPHA)
DIY(J)=FNY(DALPHA)*SIGN1
DDIX(J)=FINERX(DALPHA,DTHETA,BB)
DDIY(J)=FINERY(DALPHA,DTHETA,BB)*SIGN1
DDDIX(J)=FINERX(DALPHA,DTHETA,CC)
DDDIY(J)=FINERY(DALPHA,DTHETA,CC)*SIGN1

EIX(J)=FNX(EALPHA)
EIIY(J)=FNY(EALPHA)*SIGN1
EEIX(J)=FINERX(EALPHA,ETHETA,BB)
EEIY(J)=FINERY(EALPHA,ETHETA,BB)*SIGN1
EEEIX(J)=FINERX(EALPHA,ETHETA,CC)
EEEIY(J)=FINERY(EALPHA,ETHETA,CC)*SIGN1

FIX(J)=FNX(FALPHA)
FIY(J)=FNY(FALPHA)*SIGN1
FFIX(J)=FINERX(FALPHA,FTHETA,BB)
FFIY(J)=FINERY(FALPHA,FTHETA,BB)*SIGN1
FFFIX(J)=FINERX(FALPHA,FTHETA,CC)
FFFIY(J)=FINERY(FALPHA,FTHETA,CC)*SIGN1

GIX(J)=FNX(GALPHA)
GIY(J)=FNY(GALPHA)*SIGN1
GGIX(J)=FINERX(GALPHA,GTHETA,BB)
GGIY(J)=FINERY(GALPHA,GTHETA,BB)*SIGN1
GGGIX(J)=FINERX(GALPHA,GTHETA,CC)
GGGIY(J)=FINERY(GALPHA,GTHETA,CC)*SIGN1

HIX(J)=NIX(J-1)
HIY(J)=NIY(J-1)
HHIX(J)=NNIX(J-1)
HHIY(J)=NNIY(J-1)
```

```
GEN19210
GEN19220
GEN19230
GEN19240
GEN19250
GEN19260
GEN19270
GEN19280
GEN19290
GEN19300
GEN19310
GEN19320
GEN19330
GEN19340
GEN19350
GEN19360
GEN19370
GEN19380
GEN19390
GEN19400
GEN19410
GEN19420
GEN19430
GEN19440
GEN19450
GEN19460
GEN19470
GEN19480
GEN19490
GEN19500
GEN19510
GEN19520
GEN19530
GEN19540
GEN19550
GEN19560
GEN19570
GEN19580
GEN19590
GEN19600
GEN19610
GEN19620
GEN19630
GEN19640
GEN19650
GEN19660
GEN19670
GEN19680
GEN19690
GEN19700
GEN19710
GEN19720
GEN19730
GEN19740
GEN19750
GEN19760
GEN19770
GEN19780
GEN19790
GEN19800
GEN19810
GEN19820
GEN19830
GEN19840
```

```
IF (PHIB.GT.PI) TRANSB=-TRANSB      GEN19850
IIX(J)=XXX(R,PHIB)                  GEN19860
IIY(J)=YYY(R,PHIB)+TRANSB           GEN19870
IIIX(J)=XXX(T,PHIB)                 GEN19880
IIIIY(J)=YYY(T,PHIB)+TRANSB         GEN19890
                                      GEN19900
IF (PHIC.GT.PI) TRANSC=-TRANSC      GEN19910
JIX(J)=XXX(R,PHIC)                  GEN19920
JIY(J)=YYY(R,PHIC)+TRANSC           GEN19930
JJIX(J)=XXX(T,PHIC)                 GEN19940
JJIIY(J)=YYY(T,PHIC)+TRANSC         GEN19950
                                      GEN19960
IF (PHID.GT.PI) TRANSD=-TRANSD      GEN19970
KIX(J)=XXX(R,PHID)                  GEN19980
KIY(J)=YYY(R,PHID)+TRANSD           GEN19990
KKIX(J)=XXX(T,PHID)                 GEN20000
KKIIY(J)=YYY(T,PHID)+TRANSD         GEN20010
                                      GEN20020
IF (PHIE.GT.PI) TRANSE=-TRANSE      GEN20030
LIX(J)=XXX(R,PHIE)                  GEN20040
LIY(J)=YYY(R,PHIE)+TRANSE           GEN20050
LLIX(J)=XXX(T,PHIE)                 GEN20060
LLIIY(J)=YYY(T,PHIE)+TRANSE         GEN20070
                                      GEN20080
IF (PHIF.GT.PI) TRANSF=-TRANSF      GEN20090
MIX(J)=XXX(R,PHIF)                  GEN20100
MIY(J)=YYY(R,PHIF)+TRANSF           GEN20110
MMIX(J)=XXX(T,PHIF)                 GEN20120
MMIIY(J)=YYY(T,PHIF)+TRANSF         GEN20130
                                      GEN20140
IF (PHIG.GT.PI) TRANSG=-TRANSG      GEN20150
NIX(J)=XXX(R,PHIG)                  GEN20160
NIY(J)=YYY(R,PHIG)+TRANSG           GEN20170
NNIX(J)=XXX(T,PHIG)                 GEN20180
NNIIY(J)=YYY(T,PHIG)+TRANSG         GEN20190
                                      GEN20200
C *****                               GEN20210
C *                                     *   GEN20220
C *   IF THE RIB IS RECESSED, CALCULATE THE EXTRA POINT *   GEN20230
C *   NEEDED TO DEFINE THE RECESS OF THE RIB           *   GEN20240
C *                                                     *   GEN20250
C *****                               GEN20260
IF (RIBTYP(J).EQ.2) THEN             GEN20270
                                      GEN20280
    DRIX(J)=DIX(J)+RECES*COS(PHID)    GEN20290
    DRIY(J)=DIY(J)+RECES*SIN(PHID)    GEN20300
                                      GEN20310
    ERIX(J)=EIX(J)+RECES*COS(PHIE)    GEN20320
    ERIY(J)=EIY(J)+RECES*SIN(PHIE)    GEN20330
                                      GEN20340
    FRIX(J)=FIX(J)+RECES*COS(PHIF)    GEN20350
    FRIY(J)=FIY(J)+RECES*SIN(PHIF)    GEN20360
                                      GEN20370
    GRIX(J)=GIX(J)+RECES*COS(PHIG)    GEN20380
    GRIY(J)=GIY(J)+RECES*SIN(PHIG)    GEN20390
                                      GEN20400
ENDIF                                  GEN20410
                                      GEN20420
C *****                               GEN20430
C *                                     *   GEN20440
C *   RESET THE VALUES OF TRANS TO THEIR ORIGINAL VALUE *   GEN20450
C *                                                     *   GEN20460
C *****                               GEN20470
C *****                               GEN20480
```

```
125      NN=NN+7
          TRANSB=TRANS
          TRANSC=TRANS
          TRANSD=TRANS
          TRANSE=TRANS
          TRANSF=TRANS
          TRANSG=TRANS
          PHCHCK=0
          PHIT=PHITWO
130      CONTINUE
135      NUMBER=J-1

C      *****
C      *
C      *   THE FOLLOWING LIST OF ENTITY NUMBERING IS GIVEN SO   *
C      *   THAT IT IS EASIER FOR THE USER TO MAKE ANY CHANGES *
C      *   IN THE FEM AFTER IT IS CREATED.  THE LISTING         *
C      *   FACILLITATES EASIER LOCATION OF THE ENTITIES THAT   *
C      *   ARE TO BE CHANGED.                                   *
C      *
C      *****

140      WRITE(NO,140) NUMBER
          FORMAT('0','THE NUMBER OF RIB / CHANNEL SEGMENTS =',I3)
          WRITE(20,*) NUMBER
          WRITE(NO,*) ' '
          WRITE(NO,*) ' '

          IF(REGION(J).EQ.4) THEN

              GIY(J-1)=AIY(1)
              GGIY(J-1)=AIY(1)
              GGGIY(J-1)=AIY(1)
              NIY(J-1)=AIY(1)
              NNIY(J-1)=AIY(1)

          ENDIF

C      *****
C      *
C      *   END OF CYLINDRICAL REGION RIB/CHANNEL POINT         *
C      *   CALCULATION                                           *
C      *   (NOTE THAT THE POINT COORDINATES OF THE PORTS ARE   *
C      *   CALCULATED IN INDIVIDUAL SUBROUTINES THAT ARE       *
C      *   LOCATED AT THE END OF THIS PROGRAM)                 *
C      *
C      *****

C      *****
C      *
C      *   BEGIN GENERATION OF THE UNIVERSAL FILE               *
C      *
C      *****

C      *****
C      *
C      *   GENERATE HEADING OF UNIVERSAL FILE                   *
C      *
C      *****
```

GEN20490  
GEN20500  
GEN20510  
GEN20520  
GEN20530  
GEN20540  
GEN20550  
GEN20560  
GEN20570  
GEN20580  
GEN20590  
GEN20600  
GEN20610  
GEN20620  
GEN20630  
GEN20640  
GEN20650  
GEN20660  
GEN20670  
GEN20680  
GEN20690  
GEN20700  
GEN20710  
GEN20720  
GEN20730  
GEN20740  
GEN20750  
GEN20760  
GEN20770  
GEN20780  
GEN20790  
GEN20800  
GEN20810  
GEN20820  
GEN20830  
GEN20840  
GEN20850  
GEN20860  
GEN20870  
GEN20880  
GEN20890  
GEN20900  
GEN20910  
GEN20920  
GEN20930  
GEN20940  
GEN20950  
GEN20960  
GEN20970  
GEN20980  
GEN20990  
GEN21000  
GEN21010  
GEN21020  
GEN21030  
GEN21040  
GEN21050  
GEN21060  
GEN21070  
GEN21080  
GEN21090  
GEN21100  
GEN21110  
GEN21120

```
      WRITE(12,150)
150  FORMAT(4X,'-1')
      WRITE(12,155)
155  FORMAT(5X,'2')
      WRITE(12,160)
160  FORMAT(' IDEAS 2.2: MONITOR')
      WRITE(12,150)
      WRITE(12,150)

C *****
C *
C *   BEGIN POINT COORDINATES GENERATION OF THE INNER AND
C *   OUTER SHELLS, RIBS, AND STIFFENED CHANNELS IN
C *   UNIVERSAL FORMAT
C *
C *****

      WRITE(12,165)
165  FORMAT(4X,'25')

      IF(IEPQUE.EQ.1) THEN

          TRANS=ABS(TRANS)

          XTRAN=BB/COS(P2ODEP)
          GGIX(IEP-1)=GIX(IEP-1)+XTRAN
          GGIY(IEP-1)=GIY(IEP-1)

          XTRAN=(CC-BB)/COS(P2ODEP)
          GGGIX(IEP-1)=GGIX(IEP-1)+XTRAN
          GGGIY(IEP-1)=GGIY(IEP-1)

          PHI3=ATAN(ABS(GGIY(IEP-1)+TRANS)/GGIX(IEP-1))
          CALL PALPSL(EE,RR,PI,PHI3,TRANS,ALPHA3)
          CALL THTASL(EE,RR,PI,ALPHA3,THETA3)

          GGIX(IEP-1)=FINERX(ALPHA3,THETA3,BB)
          GGIY(IEP-1)=-FINERY(ALPHA3,THETA3,BB)

          PHI3=ATAN(ABS(GGIY(IEP-1)+TRANS)/GGIX(IEP-1))
          PHI4=P1ODEP-PHI3
          PHI5=PHI3+PHI4/3.0
          PHI6=PHI3+2.0*PHI4/3.0

          CALL PALPSL(EE,RR,PI,PHI5,TRANS,ALPHA5)
          CALL THTASL(EE,RR,PI,ALPHA5,THETA5)
          CALL PALPSL(EE,RR,PI,PHI6,TRANS,ALPHA6)
          CALL THTASL(EE,RR,PI,ALPHA6,THETA6)

          FFIX(IEP-1)=FINERX(ALPHA5,THETA5,BB)
          FFIY(IEP-1)=-FINERY(ALPHA5,THETA5,BB)

          EEIX(IEP-1)=FINERX(ALPHA6,THETA6,BB)
          EEIY(IEP-1)=-FINERY(ALPHA6,THETA6,BB)

          PHI7=ATAN(ABS(GGGIY(IEP-1)+TRANS)/GGGIX(IEP-1))
          CALL PALPSL(EE,RR,PI,PHI7,TRANS,ALPHA7)
          CALL THTASL(EE,RR,PI,ALPHA7,THETA7)

          GGGIX(IEP-1)=FINERX(ALPHA7,THETA7,CC)
          GGGIY(IEP-1)=-FINERY(ALPHA7,THETA7,CC)

          PHI7=ATAN(ABS(GGGIY(IEP-1)+TRANS)/GGGIX(IEP-1))
```

GEN21130  
GEN21140  
GEN21150  
GEN21160  
GEN21170  
GEN21180  
GEN21190  
GEN21200  
GEN21210  
GEN21220  
GEN21230  
GEN21240  
GEN21250  
GEN21260  
GEN21270  
GEN21280  
GEN21290  
GEN21300  
GEN21310  
GEN21320  
GEN21330  
GEN21340  
GEN21350  
GEN21360  
GEN21370  
GEN21380  
GEN21390  
GEN21400  
GEN21410  
GEN21420  
GEN21430  
GEN21440  
GEN21450  
GEN21460  
GEN21470  
GEN21480  
GEN21490  
GEN21500  
GEN21510  
GEN21520  
GEN21530  
GEN21540  
GEN21550  
GEN21560  
GEN21570  
GEN21580  
GEN21590  
GEN21600  
GEN21610  
GEN21620  
GEN21630  
GEN21640  
GEN21650  
GEN21660  
GEN21670  
GEN21680  
GEN21690  
GEN21700  
GEN21710  
GEN21720  
GEN21730  
GEN21740  
GEN21750  
GEN21760

```
PHI8=PIODEP-PHI7
PHI9=PHI7+PHI8/3.0
PHI10=PHI7+2.0*PHI8/3.0
```

```
CALL PALPSL (EE,RR,PI,PHI9,TRANS,ALPHA9)
CALL THTASL (EE,RR,PI,ALPHA9,THETA9)
CALL PALPSL (EE,RR,PI,PHI10,TRANS,ALPH10)
CALL THTASL (EE,RR,PI,ALPH10,THET10)
```

```
FFFIX (IEP-1)=FINERX (ALPHA9,THETA9,CC)
FFFIY (IEP-1)=-FINERY (ALPHA9,THETA9,CC)
```

```
EEEIX (IEP-1)=FINERX (ALPH10,THET10,CC)
EEEIY (IEP-1)=-FINERY (ALPH10,THET10,CC)
```

```
# CALL EXHST (EE,RR,R,PI,REXPT,TEXPT,AA,BB,CC,D,
# DEPTH,TRANS,PHI1EP,PHI2EP,P1ODEP,P2ODEP,PLNWEF,
# GIX (IEP-1),GIY (IEP-1),GGIX (IEP-1),GGIY (IEP-1),
# GGGIX (IEP-1),GGGIY (IEP-1))
```

```
TRANS=ABS (TRANS)
```

```
ENDIF
```

```
DO 185 I=1,NUMBER
```

```
  IPT=0
  ZZZ (I)=0.0
```

```
170 WRITE (12,175) IL,ICS,COLOR,BBBIX (I),BBBIY (I),ZZZ (I)
  WRITE (12,175) IL+1,ICS,COLOR,BBIX (I),BBIY (I),ZZZ (I)
  WRITE (12,175) IL+2,ICS,COLOR,BIX (I),BIY (I),ZZZ (I)
  WRITE (12,175) IL+3,ICS,COLOR,IIX (I),IIY (I),ZZZ (I)
  WRITE (12,175) IL+4,ICS,COLOR,IIIX (I),IIIIY (I),ZZZ (I)
  WRITE (12,175) IL+5,ICS,COLOR,CCCIIX (I),CCCIY (I),ZZZ (I)
  WRITE (12,175) IL+6,ICS,COLOR,CCIX (I),CCIY (I),ZZZ (I)
  WRITE (12,175) IL+7,ICS,COLOR,CIX (I),CIY (I),ZZZ (I)
  WRITE (12,175) IL+8,ICS,COLOR,JIX (I),JIY (I),ZZZ (I)
  WRITE (12,175) IL+9,ICS,COLOR,JJIX (I),JJIIY (I),ZZZ (I)
  WRITE (12,175) IL+10,ICS,COLOR,DDDIIX (I),DDDIY (I),ZZZ (I)
  WRITE (12,175) IL+11,ICS,COLOR,DDIX (I),DDIY (I),ZZZ (I)
  WRITE (12,175) IL+12,ICS,COLOR,DIX (I),DIY (I),ZZZ (I)
```

```
IF (RIBTYP (I).EQ.2) THEN
  WRITE (12,175) IL+13,ICS,COLOR,DRIX (I),DRIY (I),ZZZ (I)
  IL=IL+1
```

```
ENDIF
```

```
  WRITE (12,175) IL+13,ICS,COLOR,KIX (I),KIY (I),ZZZ (I)
  WRITE (12,175) IL+14,ICS,COLOR,KKIX (I),KKIY (I),ZZZ (I)
  WRITE (12,175) IL+15,ICS,COLOR,EEIIX (I),EEIY (I),ZZZ (I)
  WRITE (12,175) IL+16,ICS,COLOR,EEIX (I),EEIY (I),ZZZ (I)
  WRITE (12,175) IL+17,ICS,COLOR,EIX (I),EIY (I),ZZZ (I)
```

```
IF (RIBTYP (I).EQ.2) THEN
  WRITE (12,175) IL+18,ICS,COLOR,ERIX (I),ERIY (I),ZZZ (I)
  IL=IL+1
```

```
ENDIF
```

```
  WRITE (12,175) IL+18,ICS,COLOR,LIX (I),LIY (I),ZZZ (I)
  WRITE (12,175) IL+19,ICS,COLOR,LLIX (I),LLIY (I),ZZZ (I)
  WRITE (12,175) IL+20,ICS,COLOR,FFFIX (I),FFFIY (I),ZZZ (I)
```

```
GEN21770
GEN21780
GEN21790
GEN21800
GEN21810
GEN21820
GEN21830
GEN21840
GEN21850
GEN21860
GEN21870
GEN21880
GEN21890
GEN21900
GEN21910
GEN21920
GEN21930
GEN21940
GEN21950
GEN21960
GEN21970
GEN21980
GEN21990
GEN22000
GEN22010
GEN22020
GEN22030
GEN22040
GEN22050
GEN22060
GEN22070
GEN22080
GEN22090
GEN22100
GEN22110
GEN22120
GEN22130
GEN22140
GEN22150
GEN22160
GEN22170
GEN22180
GEN22190
GEN22200
GEN22210
GEN22220
GEN22230
GEN22240
GEN22250
GEN22260
GEN22270
GEN22280
GEN22290
GEN22300
GEN22310
GEN22320
GEN22330
GEN22340
GEN22350
GEN22360
GEN22370
GEN22380
GEN22390
GEN22400
```

```

WRITE(12,175) IL+21,ICS,COLOR,FFIX(I),FFIY(I),ZZZ(I)
WRITE(12,175) IL+22,ICS,COLOR,FIX(I),FIY(I),ZZZ(I)
IF(RIBTYP(I).EQ.2) THEN
  WRITE(12,175) IL+23,ICS,COLOR,FRIX(I),FRIY(I),ZZZ(I)
  IL=IL+1
ENDIF

  WRITE(12,175) IL+23,ICS,COLOR,MIX(I),MIY(I),ZZZ(I)
  WRITE(12,175) IL+24,ICS,COLOR,MMIX(I),MMIY(I),ZZZ(I)
  WRITE(12,175) IL+25,ICS,COLOR,GGGIX(I),GGGIY(I),ZZZ(I)
  WRITE(12,175) IL+26,ICS,COLOR,GGIX(I),GGIY(I),ZZZ(I)
  WRITE(12,175) IL+27,ICS,COLOR,GIX(I),GIY(I),ZZZ(I)

IF(RIBTYP(I).EQ.2) THEN
  WRITE(12,175) IL+28,ICS,COLOR,GRIX(I),GRIY(I),ZZZ(I)
  IL=IL+1
ENDIF

  WRITE(12,175) IL+28,ICS,COLOR,NIX(I),NIY(I),ZZZ(I)
  WRITE(12,175) IL+29,ICS,COLOR,NNIX(I),NNIY(I),ZZZ(I)

175  FORMAT(2I10,10X,I10,3E13.5)

  IL=IL+30
  ZZZ(I)=DEPTH

  IPT=IPT+1
  IF(IPT.LT.2) GO TO 170

    IF(CNLTYP(I).EQ.2) THEN

      RZZZ(I)=DEPTH/2.0

        IF(I.EQ.1) THEN
          WRITE(12,175) IL,ICS,COLOR,AIX(I),AIY(I),RZZZ(I)
        ELSE
          WRITE(12,175) IL,ICS,COLOR,GIX(I-1),GIY(I-1),RZZZ(I)
        ENDIF

      WRITE(12,175) IL+1,ICS,COLOR,HIX(I),HIY(I),RZZZ(I)
      WRITE(12,175) IL+2,ICS,COLOR,BIX(I),BIY(I),RZZZ(I)
      WRITE(12,175) IL+3,ICS,COLOR,IIX(I),IIY(I),RZZZ(I)
      WRITE(12,175) IL+4,ICS,COLOR,CIX(I),CIY(I),RZZZ(I)
      WRITE(12,175) IL+5,ICS,COLOR,JIX(I),JIY(I),RZZZ(I)
      WRITE(12,175) IL+6,ICS,COLOR,DIX(I),DIY(I),RZZZ(I)
      WRITE(12,175) IL+7,ICS,COLOR,KIX(I),KIY(I),RZZZ(I)

      IL=IL+8

    ENDIF

185  CONTINUE

  LASTPT=IL-1
  TEMP=LASTPT
  NPRC=LASTPT
  WRITE(NO,*) 'NUMBER OF POINTS IN RC SEGMENTS      = ',NPRC

*****
*

```

```
C *      GENERATE POINT COORDINATES OF THE EXHAUST PORT      *
C *
C *****
      IF ((IEPQUE.EQ.1).OR.(IIPQUE.EQ.1)) THEN
          IF (CCC.NE.0.0) THEN
              IEND=7
          ELSE
              IEND=6
          ENDIF
          DO 210 JJ=1,IEND
      PNTT=LASTPT
      WRITE(12,205) LASTPT+1,ICS,COLOR,X1(JJ),Y1(JJ),Z1(JJ)
      WRITE(12,205) LASTPT+2,ICS,COLOR,X2(JJ),Y2(JJ),Z2(JJ)
      WRITE(12,205) LASTPT+3,ICS,COLOR,X3(JJ),Y3(JJ),Z3(JJ)
      WRITE(12,205) LASTPT+4,ICS,COLOR,X4(JJ),Y4(JJ),Z4(JJ)
      WRITE(12,205) LASTPT+5,ICS,COLOR,X5(JJ),Y5(JJ),Z5(JJ)
      WRITE(12,205) LASTPT+6,ICS,COLOR,X6(JJ),Y6(JJ),Z6(JJ)
      WRITE(12,205) LASTPT+7,ICS,COLOR,X7(JJ),Y7(JJ),Z7(JJ)
      WRITE(12,205) LASTPT+8,ICS,COLOR,X8(JJ),Y8(JJ),Z8(JJ)
      WRITE(12,205) LASTPT+9,ICS,COLOR,X9(JJ),Y9(JJ),Z9(JJ)
      WRITE(12,205) LASTPT+10,ICS,COLOR,X10(JJ),Y10(JJ),Z10(JJ)
      WRITE(12,205) LASTPT+11,ICS,COLOR,X11(JJ),Y11(JJ),Z11(JJ)
      WRITE(12,205) LASTPT+12,ICS,COLOR,X12(JJ),Y12(JJ),Z12(JJ)
      WRITE(12,205) LASTPT+13,ICS,COLOR,X13(JJ),Y13(JJ),Z13(JJ)
      WRITE(12,205) LASTPT+14,ICS,COLOR,X14(JJ),Y14(JJ),Z14(JJ)
      WRITE(12,205) LASTPT+15,ICS,COLOR,X15(JJ),Y15(JJ),Z15(JJ)
      WRITE(12,205) LASTPT+16,ICS,COLOR,X16(JJ),Y16(JJ),Z16(JJ)
      WRITE(12,205) LASTPT+17,ICS,COLOR,X17(JJ),Y17(JJ),Z17(JJ)
      WRITE(12,205) LASTPT+18,ICS,COLOR,X18(JJ),Y18(JJ),Z18(JJ)
      WRITE(12,205) LASTPT+19,ICS,COLOR,X19(JJ),Y19(JJ),Z19(JJ)
      WRITE(12,205) LASTPT+20,ICS,COLOR,X20(JJ),Y20(JJ),Z20(JJ)
      WRITE(12,205) LASTPT+21,ICS,COLOR,X21(JJ),Y21(JJ),Z21(JJ)
      WRITE(12,205) LASTPT+22,ICS,COLOR,X22(JJ),Y22(JJ),Z22(JJ)
      WRITE(12,205) LASTPT+23,ICS,COLOR,X23(JJ),Y23(JJ),Z23(JJ)
      WRITE(12,205) LASTPT+24,ICS,COLOR,X24(JJ),Y24(JJ),Z24(JJ)
          IF ((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 190
      WRITE(12,205) LASTPT+25,ICS,COLOR,X25(JJ),Y25(JJ),Z25(JJ)
      WRITE(12,205) LASTPT+26,ICS,COLOR,X26(JJ),Y26(JJ),Z26(JJ)
      WRITE(12,205) LASTPT+27,ICS,COLOR,X27(JJ),Y27(JJ),Z27(JJ)
      WRITE(12,205) LASTPT+28,ICS,COLOR,X28(JJ),Y28(JJ),Z28(JJ)
      WRITE(12,205) LASTPT+29,ICS,COLOR,X29(JJ),Y29(JJ),Z29(JJ)
      WRITE(12,205) LASTPT+30,ICS,COLOR,X30(JJ),Y30(JJ),Z30(JJ)
      LASTPT=LASTPT+6
190  LASTPT=LASTPT-6
          IF ((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 195
      WRITE(12,205) LASTPT+31,ICS,COLOR,X31(JJ),Y31(JJ),Z31(JJ)
      WRITE(12,205) LASTPT+32,ICS,COLOR,X32(JJ),Y32(JJ),Z32(JJ)
      WRITE(12,205) LASTPT+33,ICS,COLOR,X33(JJ),Y33(JJ),Z33(JJ)
      WRITE(12,205) LASTPT+34,ICS,COLOR,X34(JJ),Y34(JJ),Z34(JJ)
      LASTPT=LASTPT+4
195  LASTPT=LASTPT-4
```

```
      IF ((JJ.EQ.3).OR.(JJ.EQ.4).OR.(JJ.EQ.5).
#       OR.(JJ.EQ.6).OR.(JJ.EQ.7)) GO TO 200
      WRITE(12,205) LASTPT+35, ICS, COLOR, X35(JJ), Y35(JJ), Z35(JJ)
      WRITE(12,205) LASTPT+36, ICS, COLOR, X36(JJ), Y36(JJ), Z36(JJ)
      WRITE(12,205) LASTPT+37, ICS, COLOR, X37(JJ), Y37(JJ), Z37(JJ)
      WRITE(12,205) LASTPT+38, ICS, COLOR, X38(JJ), Y38(JJ), Z38(JJ)
      WRITE(12,205) LASTPT+39, ICS, COLOR, X39(JJ), Y39(JJ), Z39(JJ)
      WRITE(12,205) LASTPT+40, ICS, COLOR, X40(JJ), Y40(JJ), Z40(JJ)
      WRITE(12,205) LASTPT+41, ICS, COLOR, X41(JJ), Y41(JJ), Z41(JJ)
      WRITE(12,205) LASTPT+42, ICS, COLOR, X42(JJ), Y42(JJ), Z42(JJ)
      WRITE(12,205) LASTPT+43, ICS, COLOR, X43(JJ), Y43(JJ), Z43(JJ)
      WRITE(12,205) LASTPT+44, ICS, COLOR, X44(JJ), Y44(JJ), Z44(JJ)
      WRITE(12,205) LASTPT+45, ICS, COLOR, X45(JJ), Y45(JJ), Z45(JJ)
      WRITE(12,205) LASTPT+46, ICS, COLOR, X46(JJ), Y46(JJ), Z46(JJ)
      WRITE(12,205) LASTPT+47, ICS, COLOR, X47(JJ), Y47(JJ), Z47(JJ)
      WRITE(12,205) LASTPT+48, ICS, COLOR, X48(JJ), Y48(JJ), Z48(JJ)
      WRITE(12,205) LASTPT+49, ICS, COLOR, X49(JJ), Y49(JJ), Z49(JJ)
      LASTPT=LASTPT+15
200    LASTPT=LASTPT-15
205    FORMAT(2I10,10X,I10,3E13.5)
      LASTPT=LASTPT+49
      PNTB=LASTPT
      NPTEP(JJ)=PNTB-PNTT
210          CONTINUE
      ENDIF
      NPEP=LASTPT-NPRC
      WRITE(NO,*) 'NUMBER OF POINTS IN EXHAUST PORT = ',NPEP
C *****
C *
C *      GENERATE POINT COORDINATES OF THE INTAKE PORT
C *
C *****
      IF (IIPQUE.EQ.1) THEN
      CALL INTKE (EE,RR,R,PI,RINPT,TINPT,AA,BB,CC,D,DEPTH,
#      TRANS,PHI1IP,PHI2IP,P2ODIP,P1NWIP,LINPT,WINPT,NO)
      DO 225 JJ=1,IEND
      PNTT=LASTPT
      WRITE(12,205) LASTPT+1, ICS, COLOR, X1(JJ), Y1(JJ), Z1(JJ)
      WRITE(12,205) LASTPT+2, ICS, COLOR, X2(JJ), Y2(JJ), Z2(JJ)
      WRITE(12,205) LASTPT+3, ICS, COLOR, X3(JJ), Y3(JJ), Z3(JJ)
      WRITE(12,205) LASTPT+4, ICS, COLOR, X4(JJ), Y4(JJ), Z4(JJ)
      WRITE(12,205) LASTPT+5, ICS, COLOR, X5(JJ), Y5(JJ), Z5(JJ)
      WRITE(12,205) LASTPT+6, ICS, COLOR, X6(JJ), Y6(JJ), Z6(JJ)
      WRITE(12,205) LASTPT+7, ICS, COLOR, X7(JJ), Y7(JJ), Z7(JJ)
      WRITE(12,205) LASTPT+8, ICS, COLOR, X8(JJ), Y8(JJ), Z8(JJ)
      WRITE(12,205) LASTPT+9, ICS, COLOR, X9(JJ), Y9(JJ), Z9(JJ)
      WRITE(12,205) LASTPT+10, ICS, COLOR, X10(JJ), Y10(JJ), Z10(JJ)
      WRITE(12,205) LASTPT+11, ICS, COLOR, X11(JJ), Y11(JJ), Z11(JJ)
      WRITE(12,205) LASTPT+12, ICS, COLOR, X12(JJ), Y12(JJ), Z12(JJ)
      WRITE(12,205) LASTPT+13, ICS, COLOR, X13(JJ), Y13(JJ), Z13(JJ)
      WRITE(12,205) LASTPT+14, ICS, COLOR, X14(JJ), Y14(JJ), Z14(JJ)
      WRITE(12,205) LASTPT+15, ICS, COLOR, X15(JJ), Y15(JJ), Z15(JJ)
      WRITE(12,205) LASTPT+16, ICS, COLOR, X16(JJ), Y16(JJ), Z16(JJ)
```



```
WRITE(12,205) LASTPT+17,ICS,COLOR,X17(JJ),Y17(JJ),Z17(JJ)
WRITE(12,205) LASTPT+18,ICS,COLOR,X18(JJ),Y18(JJ),Z18(JJ)
WRITE(12,205) LASTPT+19,ICS,COLOR,X19(JJ),Y19(JJ),Z19(JJ)
WRITE(12,205) LASTPT+20,ICS,COLOR,X20(JJ),Y20(JJ),Z20(JJ)
WRITE(12,205) LASTPT+21,ICS,COLOR,X21(JJ),Y21(JJ),Z21(JJ)
WRITE(12,205) LASTPT+22,ICS,COLOR,X22(JJ),Y22(JJ),Z22(JJ)
WRITE(12,205) LASTPT+23,ICS,COLOR,X23(JJ),Y23(JJ),Z23(JJ)
WRITE(12,205) LASTPT+24,ICS,COLOR,X24(JJ),Y24(JJ),Z24(JJ)
WRITE(12,205) LASTPT+25,ICS,COLOR,X25(JJ),Y25(JJ),Z25(JJ)
WRITE(12,205) LASTPT+26,ICS,COLOR,X26(JJ),Y26(JJ),Z26(JJ)
```

```
IF((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 215
```

```
WRITE(12,205) LASTPT+27,ICS,COLOR,X27(JJ),Y27(JJ),Z27(JJ)
WRITE(12,205) LASTPT+28,ICS,COLOR,X28(JJ),Y28(JJ),Z28(JJ)
WRITE(12,205) LASTPT+29,ICS,COLOR,X29(JJ),Y29(JJ),Z29(JJ)
WRITE(12,205) LASTPT+30,ICS,COLOR,X30(JJ),Y30(JJ),Z30(JJ)
WRITE(12,205) LASTPT+31,ICS,COLOR,X31(JJ),Y31(JJ),Z31(JJ)
WRITE(12,205) LASTPT+32,ICS,COLOR,X32(JJ),Y32(JJ),Z32(JJ)
WRITE(12,205) LASTPT+33,ICS,COLOR,X33(JJ),Y33(JJ),Z33(JJ)
WRITE(12,205) LASTPT+34,ICS,COLOR,X34(JJ),Y34(JJ),Z34(JJ)
WRITE(12,205) LASTPT+35,ICS,COLOR,X35(JJ),Y35(JJ),Z35(JJ)
WRITE(12,205) LASTPT+36,ICS,COLOR,X36(JJ),Y36(JJ),Z36(JJ)
WRITE(12,205) LASTPT+37,ICS,COLOR,X37(JJ),Y37(JJ),Z37(JJ)
WRITE(12,205) LASTPT+38,ICS,COLOR,X38(JJ),Y38(JJ),Z38(JJ)
LASTPT=LASTPT+12
```

```
215 LASTPT=LASTPT-12
```

```
IF((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 220
```

```
WRITE(12,205) LASTPT+39,ICS,COLOR,X39(JJ),Y39(JJ),Z39(JJ)
WRITE(12,205) LASTPT+40,ICS,COLOR,X40(JJ),Y40(JJ),Z40(JJ)
WRITE(12,205) LASTPT+41,ICS,COLOR,X41(JJ),Y41(JJ),Z41(JJ)
WRITE(12,205) LASTPT+42,ICS,COLOR,X42(JJ),Y42(JJ),Z42(JJ)
WRITE(12,205) LASTPT+43,ICS,COLOR,X43(JJ),Y43(JJ),Z43(JJ)
WRITE(12,205) LASTPT+44,ICS,COLOR,X44(JJ),Y44(JJ),Z44(JJ)
WRITE(12,205) LASTPT+45,ICS,COLOR,X45(JJ),Y45(JJ),Z45(JJ)
WRITE(12,205) LASTPT+46,ICS,COLOR,X46(JJ),Y46(JJ),Z46(JJ)
WRITE(12,205) LASTPT+47,ICS,COLOR,X47(JJ),Y47(JJ),Z47(JJ)
WRITE(12,205) LASTPT+48,ICS,COLOR,X48(JJ),Y48(JJ),Z48(JJ)
WRITE(12,205) LASTPT+49,ICS,COLOR,X49(JJ),Y49(JJ),Z49(JJ)
WRITE(12,205) LASTPT+50,ICS,COLOR,X50(JJ),Y50(JJ),Z50(JJ)
WRITE(12,205) LASTPT+51,ICS,COLOR,X51(JJ),Y51(JJ),Z51(JJ)
WRITE(12,205) LASTPT+52,ICS,COLOR,X52(JJ),Y52(JJ),Z52(JJ)
WRITE(12,205) LASTPT+53,ICS,COLOR,X53(JJ),Y53(JJ),Z53(JJ)
LASTPT=LASTPT+15
```

```
220 LASTPT=LASTPT-15
```

```
LASTPT=LASTPT+53
PNTB=LASTPT
NPTIP(JJ)=PNTB-PNTT
```

```
225 CONTINUE
```

```
ENDIF
```

```
NPTTL=LASTPT
```

```
NPIP=LASTPT-NPRC-NPEP
```

```
WRITE(NO,*) 'NUMBER OF POINTS IN INTAKE PORT = ',NPIP
```

```
*****
```

GEN24330  
GEN24340  
GEN24350  
GEN24360  
GEN24370  
GEN24380  
GEN24390  
GEN24400  
GEN24410  
GEN24420  
GEN24430  
GEN24440  
GEN24450  
GEN24460  
GEN24470  
GEN24480  
GEN24490  
GEN24500  
GEN24510  
GEN24520  
GEN24530  
GEN24540  
GEN24550  
GEN24560  
GEN24570  
GEN24580  
GEN24590  
GEN24600  
GEN24610  
GEN24620  
GEN24630  
GEN24640  
GEN24650  
GEN24660  
GEN24670  
GEN24680  
GEN24690  
GEN24700  
GEN24710  
GEN24720  
GEN24730  
GEN24740  
GEN24750  
GEN24760  
GEN24770  
GEN24780  
GEN24790  
GEN24800  
GEN24810  
GEN24820  
GEN24830  
GEN24840  
GEN24850  
GEN24860  
GEN24870  
GEN24880  
GEN24890  
GEN24900  
GEN24910  
GEN24920  
GEN24930  
GEN24940  
GEN24950  
GEN24960

```
C *
C *      GENERATE POINT COORDINATES OF THE SPARK PLUG      *
C *
C *****
      IF (ISP.GT.0) THEN
        DO 235 J=1,ISP
          CALL SPRKPG (EE,RR,R,PI,Y1SP(J),Y2SP(J),PHI1SP(J),
#             PHI2SP(J),RSP(J),IEND,AA,BB,CC,D,DEPTH,
#             TRANS,REGION(ISPRK(J)),ICLK,NO)
          DO 230 JJ=1,IEND
            PNTT=LASTPT
            WRITE(12,205) LASTPT+1,ICS,COLOR,X1(JJ),Y1(JJ),Z1(JJ)
            WRITE(12,205) LASTPT+2,ICS,COLOR,X2(JJ),Y2(JJ),Z2(JJ)
            WRITE(12,205) LASTPT+3,ICS,COLOR,X3(JJ),Y3(JJ),Z3(JJ)
            WRITE(12,205) LASTPT+4,ICS,COLOR,X4(JJ),Y4(JJ),Z4(JJ)
            WRITE(12,205) LASTPT+5,ICS,COLOR,X5(JJ),Y5(JJ),Z5(JJ)
            WRITE(12,205) LASTPT+6,ICS,COLOR,X6(JJ),Y6(JJ),Z6(JJ)
            WRITE(12,205) LASTPT+7,ICS,COLOR,X7(JJ),Y7(JJ),Z7(JJ)
            WRITE(12,205) LASTPT+8,ICS,COLOR,X8(JJ),Y8(JJ),Z8(JJ)
            WRITE(12,205) LASTPT+9,ICS,COLOR,X9(JJ),Y9(JJ),Z9(JJ)
            WRITE(12,205) LASTPT+10,ICS,COLOR,X10(JJ),Y10(JJ),Z10(JJ)
            WRITE(12,205) LASTPT+11,ICS,COLOR,X11(JJ),Y11(JJ),Z11(JJ)
            WRITE(12,205) LASTPT+12,ICS,COLOR,X12(JJ),Y12(JJ),Z12(JJ)
            WRITE(12,205) LASTPT+13,ICS,COLOR,X13(JJ),Y13(JJ),Z13(JJ)
            WRITE(12,205) LASTPT+14,ICS,COLOR,X14(JJ),Y14(JJ),Z14(JJ)
            WRITE(12,205) LASTPT+15,ICS,COLOR,X15(JJ),Y15(JJ),Z15(JJ)
            WRITE(12,205) LASTPT+16,ICS,COLOR,X16(JJ),Y16(JJ),Z16(JJ)
            WRITE(12,205) LASTPT+17,ICS,COLOR,X17(JJ),Y17(JJ),Z17(JJ)
            WRITE(12,205) LASTPT+18,ICS,COLOR,X18(JJ),Y18(JJ),Z18(JJ)
            WRITE(12,205) LASTPT+19,ICS,COLOR,X19(JJ),Y19(JJ),Z19(JJ)
            WRITE(12,205) LASTPT+20,ICS,COLOR,X20(JJ),Y20(JJ),Z20(JJ)
            WRITE(12,205) LASTPT+21,ICS,COLOR,X21(JJ),Y21(JJ),Z21(JJ)
            WRITE(12,205) LASTPT+22,ICS,COLOR,X22(JJ),Y22(JJ),Z22(JJ)
            WRITE(12,205) LASTPT+23,ICS,COLOR,X23(JJ),Y23(JJ),Z23(JJ)
            WRITE(12,205) LASTPT+24,ICS,COLOR,X24(JJ),Y24(JJ),Z24(JJ)
            WRITE(12,205) LASTPT+25,ICS,COLOR,X25(JJ),Y25(JJ),Z25(JJ)
            WRITE(12,205) LASTPT+26,ICS,COLOR,X26(JJ),Y26(JJ),Z26(JJ)
            WRITE(12,205) LASTPT+27,ICS,COLOR,X27(JJ),Y27(JJ),Z27(JJ)
            WRITE(12,205) LASTPT+28,ICS,COLOR,X28(JJ),Y28(JJ),Z28(JJ)
            WRITE(12,205) LASTPT+29,ICS,COLOR,X29(JJ),Y29(JJ),Z29(JJ)
            WRITE(12,205) LASTPT+30,ICS,COLOR,X30(JJ),Y30(JJ),Z30(JJ)
            LASTPT=LASTPT+30
            PNTB=LASTPT
            NPTSP(JJ)=PNTB-PNTT
230          CONTINUE
235          CONTINUE
          NPSP=LASTPT-NPRC-NPEP-NPIP
          WRITE(NO,*) 'NUMBER OF POINTS IN SPARK PLUG(S) = ',NPSP
        ENDIF
        WRITE(NO,*) 'TOTAL NUMBER OF POINTS = ',LASTPT
        WRITE(NO,*) ' '
        WRITE(NO,*) ' '
```

```

WRITE(NO,*)'POINT LABELS OF RC SEGMENTS ARE          1 TO',NPRC GEN25610
WRITE(NO,*)'POINT LABELS OF EXHST PORT ARE',NPRC+1,'TO',NPEP+NPRC GEN25620
WRITE(NO,*)'POINT LABELS OF INTKE PORT ARE',NPRC+NPEP+1,'TO',NPIP+GEN25630
#NPRC+NPEP GEN25640
WRITE(NO,*)'POINT LABELS OF SPRK PLUGS ARE',NPRC+NPEP+NPIP+1,'TO',GEN25650
#NPSP+NPRC+NPEP+NPIP GEN25660
WRITE(NO,*) ' ' GEN25670
WRITE(NO,*) ' ' GEN25680

```

```

      LASTPT=TEMP
      TEMP1=TEMP
WRITE(12,150)
WRITE(12,150)

```

```

C *****
C *
C *          END GENERATION OF THE POINT COORDINATES
C *
C *****

```

```

C *****
C *
C *          BEGIN LINE GENERATION OF THE INNER AND OUTER SHELLS,
C *          RIBS, AND STIFFENED CHANNELS IN UNIVERSAL FORMAT
C *
C *****

```

```

      CALL COIN (NO)
      CALL FORGEN (COLOR,SOLID,DASH,GRADE,NUMBER,TEMP,TEMP1,
#          IEND, LASTPT, IEPQUE, IIPQUE, SDC, NO, ICONTI)

      CALL MNE (NO)
      CALL GO (ICONTI)

```

```

      STOP
      END

```

```

C *****
C *
C *          BEGIN SUBROUTINE LIBRARY
C *
C *****

```

```

      SUBROUTINE FORGEN (COLOR,SOLID,DASH,GRADE,NUMBER,TEMP,TEMP1,
#          IEND, LASTPT, IEPQUE, IIPQUE, SDC, NO, ICONTI)

```

```

      DIMENSION NLNEP(7),NLNEP1(7),NLNIP(7),NLNIP1(7),NSREP(7),NSRIP(7)
#          ,NLNIP2(7),NSIP2(7),NLNSP(7),NSRSP(7),NSIP3(7)

```

```

      COMMON / MAIN2 / CNLTYP(100),RIBTYP(100),NPTEP(7),NPTIP(7)
#          ,NPTSP(7),NPTTL,ISP,IVOLR(25),IVOLC(25),TCHNL,CCC

```

```

      INTEGER PT1,PT2,PT3,PT4,RIBTYP,CNLTYP,SDC
#          ,SOLID,COLOR,START,STOP,COUNT,SPLINE
#          ,SPENTC,LNENTC,DASH,GRADE,SURFAC,ED1,ED2,ED3,ED4
#          ,VOLUME,SUR1,SUR2,SUR3,SUR4,SUR5,SUR6,SLSLVR
#          ,SURT,SURB,EDGE,TEMPID,TEMP11

```

```

      WRITE(12,240)
240  FORMAT(4X,'26')
245  FORMAT(4X,'-1')

```

```

      LINE=1
      START=11

```

GEN25610  
 GEN25620  
 GEN25630  
 GEN25640  
 GEN25650  
 GEN25660  
 GEN25670  
 GEN25680  
 GEN25690  
 GEN25700  
 GEN25710  
 GEN25720  
 GEN25730  
 GEN25740  
 GEN25750  
 GEN25760  
 GEN25770  
 GEN25780  
 GEN25790  
 GEN25800  
 GEN25810  
 GEN25820  
 GEN25830  
 GEN25840  
 GEN25850  
 GEN25860  
 GEN25870  
 GEN25880  
 GEN25890  
 GEN25900  
 GEN25910  
 GEN25920  
 GEN25930  
 GEN25940  
 GEN25950  
 GEN25960  
 GEN25970  
 GEN25980  
 GEN25990  
 GEN26000  
 GEN26010  
 GEN26020  
 GEN26030  
 GEN26040  
 GEN26050  
 GEN26060  
 GEN26070  
 GEN26080  
 GEN26090  
 GEN26100  
 GEN26110  
 GEN26120  
 GEN26130  
 GEN26140  
 GEN26150  
 GEN26160  
 GEN26170  
 GEN26180  
 GEN26190  
 GEN26200  
 GEN26210  
 GEN26220  
 GEN26230  
 GEN26240

```
STOP=12
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
250 FORMAT(5I10)

DO 285 I=1,NUMBER

    COUNT=0

    IF(RIBTYP(I).EQ.1) THEN
255        DO 260 J=1,3

                START=START+1
                STOP=STOP+1
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP

260        CONTINUE

                START=START-3
                STOP=STOP+26
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP

                DO 265 J=1,4

                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP

265        CONTINUE

                START=START+26
                STOP=STOP-3
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP

                DO 270 J=1,3

                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP

270        CONTINUE

                COUNT=COUNT+1
                IF(COUNT.EQ.1) THEN

                        START=START-18
                        STOP=STOP-18
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                        GO TO 255

                ENDIF

                ELSE IF(RIBTYP(I).EQ.2) THEN

                        COUNT=0

275        START=START+1
                STOP=STOP+1
```

GEN26250  
GEN26260  
GEN26270  
GEN26280  
GEN26290  
GEN26300  
GEN26310  
GEN26320  
GEN26330  
GEN26340  
GEN26350  
GEN26360  
GEN26370  
GEN26380  
GEN26390  
GEN26400  
GEN26410  
GEN26420  
GEN26430  
GEN26440  
GEN26450  
GEN26460  
GEN26470  
GEN26480  
GEN26490  
GEN26500  
GEN26510  
GEN26520  
GEN26530  
GEN26540  
GEN26550  
GEN26560  
GEN26570  
GEN26580  
GEN26590  
GEN26600  
GEN26610  
GEN26620  
GEN26630  
GEN26640  
GEN26650  
GEN26660  
GEN26670  
GEN26680  
GEN26690  
GEN26700  
GEN26710  
GEN26720  
GEN26730  
GEN26740  
GEN26750  
GEN26760  
GEN26770  
GEN26780  
GEN26790  
GEN26800  
GEN26810  
GEN26820  
GEN26830  
GEN26840  
GEN26850  
GEN26860  
GEN26870  
GEN26880

```
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START+2
STOP=STOP+2
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START-4
STOP=STOP+29
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

DO 280 J=1,5

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP

280 CONTINUE

START=START+29
STOP=STOP-4
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START+2
STOP=STOP+2
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

COUNT=COUNT+1
IF (COUNT.EQ.1) THEN

    START=START-20
    STOP=STOP-20
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP
    GO TO 275

ENDIF
ENDIF

IF (RIBTYP(I).EQ.1) THEN

    IF (CNLTYP(I).EQ.1) THEN

        IF (I.EQ.NUMBER) GO TO 285
```

```
GEN26890
GEN26900
GEN26910
GEN26920
GEN26930
GEN26940
GEN26950
GEN26960
GEN26970
GEN26980
GEN26990
GEN27000
GEN27010
GEN27020
GEN27030
GEN27040
GEN27050
GEN27060
GEN27070
GEN27080
GEN27090
GEN27100
GEN27110
GEN27120
GEN27130
GEN27140
GEN27150
GEN27160
GEN27170
GEN27180
GEN27190
GEN27200
GEN27210
GEN27220
GEN27230
GEN27240
GEN27250
GEN27260
GEN27270
GEN27280
GEN27290
GEN27300
GEN27310
GEN27320
GEN27330
GEN27340
GEN27350
GEN27360
GEN27370
GEN27380
GEN27390
GEN27400
GEN27410
GEN27420
GEN27430
GEN27440
GEN27450
GEN27460
GEN27470
GEN27480
GEN27490
GEN27500
GEN27510
GEN27520
```

```

      START=START+12
      STOP=STOP+12
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      ELSE IF (CNLTYP(I).EQ.2) THEN

      START=START+2
      STOP=STOP+2
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      START=START+6
      STOP=STOP+6
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      IF (I.EQ.NUMBER) GO TO 285
      START=START+12
      STOP=STOP+12
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP
    ENDIF

```

```

  ELSE IF (RIBTYP(I).EQ.2) THEN

```

```

    IF (I.EQ.NUMBER) GO TO 285
    START=START+12
    STOP=STOP+12
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP
  ENDIF

```

```

285  CONTINUE

```

```

  NLRC=LINE
  WRITE(NO,*) 'NUMBER OF LINES IN RC SEGMENTS = ',NLRC

```

```

C *****
C *
C *
C *
C *
C *****

```

```

  LINETP=LINE
  IF (IEPQUE.EQ.1) THEN

```

```

    DO 330 JJ=1,IEND

```

```

      LNT=LINE

```

```

        START=TEMP+1
        STOP=TEMP+2
        LINE=LINE+1
        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
        ICK=1

```

```

        DO 290 J=1,22

```

```

          START=START+1
          STOP=STOP+1
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

```

```

GEN27530
GEN27540
GEN27550
GEN27560
GEN27570
GEN27580
GEN27590
GEN27600
GEN27610
GEN27620
GEN27630
GEN27640
GEN27650
GEN27660
GEN27670
GEN27680
GEN27690
GEN27700
GEN27710
GEN27720
GEN27730
GEN27740
GEN27750
GEN27760
GEN27770
GEN27780
GEN27790
GEN27800
GEN27810
GEN27820
GEN27830
GEN27840
GEN27850
GEN27860
GEN27870
GEN27880
GEN27890
GEN27900
GEN27910
GEN27920
GEN27930
GEN27940
GEN27950
GEN27960
GEN27970
GEN27980
GEN27990
GEN28000
GEN28010
GEN28020
GEN28030
GEN28040
GEN28050
GEN28060
GEN28070
GEN28080
GEN28090
GEN28100
GEN28110
GEN28120
GEN28130
GEN28140
GEN28150
GEN28160

```

```
                IF ((J.EQ.10).OR.(J.EQ.22)) THEN
                        START=STOP
                        STOP=STOP-11
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                        START=START-1
                        STOP=STOP+11
                ENDIF
290      CONTINUE
                START=TEMP+1
                STOP=TEMP+14
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                ICK1=1
                DO 295 J=1,10-
                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
295      CONTINUE
                IF ((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 310
                START=TEMP+15
                STOP=TEMP+25
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                DO 300 J=1,8
                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
300      CONTINUE
                        IF (J.EQ.4) START=START+1
                START=TEMP+13
                STOP=TEMP+34
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                START=TEMP+25
                STOP=TEMP+26
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                DO 305 J=1,7
                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                IF (J.EQ.3) THEN
```

GEN28170  
GEN28180  
GEN28190  
GEN28200  
GEN28210  
GEN28220  
GEN28230  
GEN28240  
GEN28250  
GEN28260  
GEN28270  
GEN28280  
GEN28290  
GEN28300  
GEN28310  
GEN28320  
GEN28330  
GEN28340  
GEN28350  
GEN28360  
GEN28370  
GEN28380  
GEN28390  
GEN28400  
GEN28410  
GEN28420  
GEN28430  
GEN28440  
GEN28450  
GEN28460  
GEN28470  
GEN28480  
GEN28490  
GEN28500  
GEN28510  
GEN28520  
GEN28530  
GEN28540  
GEN28550  
GEN28560  
GEN28570  
GEN28580  
GEN28590  
GEN28600  
GEN28610  
GEN28620  
GEN28630  
GEN28640  
GEN28650  
GEN28660  
GEN28670  
GEN28680  
GEN28690  
GEN28700  
GEN28710  
GEN28720  
GEN28730  
GEN28740  
GEN28750  
GEN28760  
GEN28770  
GEN28780  
GEN28790  
GEN28800

```

                                START=START+1
                                STOP=STOP+1
                                ENDIF
305      CONTINUE
310      IF(JJ.EQ.IEND) GO TO 325

          START=TEMP+1
          STOP=TEMP+1+NPTEP(JJ)
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

          DO 315 J=1,23

                START=START+1
                STOP=STOP+1
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
315      CONTINUE

          IF((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 325

          START=TEMP+25
          STOP=TEMP+25+NPTEP(JJ)
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

          DO 320 J=1,9

                START=START+1
                STOP=STOP+1
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
320      CONTINUE
325      TEMP=TEMP+NPTEP(JJ)
          LNB=LINE
          NLNEP(JJ)=LNB-LNT
330      CONTINUE

          DO 355 JJ=1,2

                LNT=LINE
                START=TEMP1+35
                STOP=TEMP1+36
                LINE=LINE+1
                WRITE(12,250) LINE,COLOR,SOLID,START,STOP

                DO 335 J=1,11

                        START=START+1
                        STOP=STOP+1
                        LINE=LINE+1
                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP

                        IF((J.EQ.3).OR.(J.EQ.7)) THEN

                                START=START+1
                                STOP=STOP+1
```

```

GEN28810
GEN28820
GEN28830
GEN28840
GEN28850
GEN28860
GEN28870
GEN28880
GEN28890
GEN28900
GEN28910
GEN28920
GEN28930
GEN28940
GEN28950
GEN28960
GEN28970
GEN28980
GEN28990
GEN29000
GEN29010
GEN29020
GEN29030
GEN29040
GEN29050
GEN29060
GEN29070
GEN29080
GEN29090
GEN29100
GEN29110
GEN29120
GEN29130
GEN29140
GEN29150
GEN29160
GEN29170
GEN29180
GEN29190
GEN29200
GEN29210
GEN29220
GEN29230
GEN29240
GEN29250
GEN29260
GEN29270
GEN29280
GEN29290
GEN29300
GEN29310
GEN29320
GEN29330
GEN29340
GEN29350
GEN29360
GEN29370
GEN29380
GEN29390
GEN29400
GEN29410
GEN29420
GEN29430
GEN29440
```



ENDIF

335

CONTINUE

```
START=TEMP1+35
STOP=TEMP1+40
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

DO 340 J=1,9

```
START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

340

CONTINUE

```
START=TEMP1+13
STOP=TEMP1+36
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

DO 345 J=1,2

```
START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

345

CONTINUE

```
START=TEMP1+25
STOP=TEMP1+39
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

```
START=TEMP1+34
STOP=TEMP1+35
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

IF(JJ.EQ.2) GO TO 355

```
START=TEMP1+35
STOP=TEMP1+35+NPTEP(JJ)
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

DO 350 J=1,14

```
START=START+1
STOP=STOP+1
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

350

CONTINUE

```
TEMP1=TEMP1+NPTEP(JJ)
LNB=LINE
NLNEP1(JJ)=LNB-LNT
```

355

CONTINUE

GEN29450  
GEN29460  
GEN29470  
GEN29480  
GEN29490  
GEN29500  
GEN29510  
GEN29520  
GEN29530  
GEN29540  
GEN29550  
GEN29560  
GEN29570  
GEN29580  
GEN29590  
GEN29600  
GEN29610  
GEN29620  
GEN29630  
GEN29640  
GEN29650  
GEN29660  
GEN29670  
GEN29680  
GEN29690  
GEN29700  
GEN29710  
GEN29720  
GEN29730  
GEN29740  
GEN29750  
GEN29760  
GEN29770  
GEN29780  
GEN29790  
GEN29800  
GEN29810  
GEN29820  
GEN29830  
GEN29840  
GEN29850  
GEN29860  
GEN29870  
GEN29880  
GEN29890  
GEN29900  
GEN29910  
GEN29920  
GEN29930  
GEN29940  
GEN29950  
GEN29960  
GEN29970  
GEN29980  
GEN29990  
GEN30000  
GEN30010  
GEN30020  
GEN30030  
GEN30040  
GEN30050  
GEN30060  
GEN30070  
GEN30080

```

      ENDIF
      TEMP1=TEMP

      NLEP=LINE-NLRC
      WRITE(NO,*) 'NUMBER OF LINES IN EXHAUST PORT = ',NLEP

C *****
C *
C *      BEGIN LINE GENERATION OF THE INTAKE PORT
C *
C *****

      IF (IIPQUE.EQ.1) THEN

        DO 400 JJ=1,IEND

          LNT=LINE
          START=TEMP+1
          STOP=TEMP+2
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP
          ICK=1

          DO 360 J=1,25

            START=START+1
            STOP=STOP+1
            LINE=LINE+1
            WRITE(12,250) LINE,COLOR,SOLID,START,STOP

            IF (J.EQ.10) THEN

              START=STOP
              STOP=STOP-11
              LINE=LINE+1
              WRITE(12,250) LINE,COLOR,SOLID,START,STOP
              START=START-1
              STOP=STOP+11

            ENDIF

            IF (J.EQ.24) STOP=STOP-14

          CONTINUE

          START=TEMP+1
          STOP=TEMP+14
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP
          ICK1=1

          DO 365 J=1,12

            START=START+1
            STOP=STOP+1
            LINE=LINE+1
            WRITE(12,250) LINE,COLOR,SOLID,START,STOP

            IF (J.EQ.1) START=START-1

          CONTINUE

        IF ((JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 380

```

```

GEN30090
GEN30100
GEN30110
GEN30120
GEN30130
GEN30140
GEN30150
GEN30160
GEN30170
GEN30180
GEN30190
GEN30200
GEN30210
GEN30220
GEN30230
GEN30240
GEN30250
GEN30260
GEN30270
GEN30280
GEN30290
GEN30300
GEN30310
GEN30320
GEN30330
GEN30340
GEN30350
GEN30360
GEN30370
GEN30380
GEN30390
GEN30400
GEN30410
GEN30420
GEN30430
GEN30440
GEN30450
GEN30460
GEN30470
GEN30480
GEN30490
GEN30500
GEN30510
GEN30520
GEN30530
GEN30540
GEN30550
GEN30560
GEN30570
GEN30580
GEN30590
GEN30600
GEN30610
GEN30620
GEN30630
GEN30640
GEN30650
GEN30660
GEN30670
GEN30680
GEN30690
GEN30700
GEN30710
GEN30720

```

```
START=TEMP+15
STOP=TEMP+27
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

DO 370 J=1,11

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP

        IF (J.EQ.5) START=START+1
        IF (J.EQ.10) START=START-14

370    CONTINUE

START=TEMP+27
STOP=TEMP+28
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

DO 375 J=1,9

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP

        IF (J.EQ.4) START=START+1
        IF (J.EQ.4) STOP=STOP+1

375    CONTINUE

380    IF(JJ.EQ.IEND) GO TO 395

START=TEMP+1
STOP=TEMP+1+NPTIP(JJ)
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

DO 385 J=1,25

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP

385    CONTINUE

IF((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)) GO TO 395

START=TEMP+27
STOP=TEMP+27+NPTIP(JJ)
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

DO 390 J=1,11

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

GEN30730  
GEN30740  
GEN30750  
GEN30760  
GEN30770  
GEN30780  
GEN30790  
GEN30800  
GEN30810  
GEN30820  
GEN30830  
GEN30840  
GEN30850  
GEN30860  
GEN30870  
GEN30880  
GEN30890  
GEN30900  
GEN30910  
GEN30920  
GEN30930  
GEN30940  
GEN30950  
GEN30960  
GEN30970  
GEN30980  
GEN30990  
GEN31000  
GEN31010  
GEN31020  
GEN31030  
GEN31040  
GEN31050  
GEN31060  
GEN31070  
GEN31080  
GEN31090  
GEN31100  
GEN31110  
GEN31120  
GEN31130  
GEN31140  
GEN31150  
GEN31160  
GEN31170  
GEN31180  
GEN31190  
GEN31200  
GEN31210  
GEN31220  
GEN31230  
GEN31240  
GEN31250  
GEN31260  
GEN31270  
GEN31280  
GEN31290  
GEN31300  
GEN31310  
GEN31320  
GEN31330  
GEN31340  
GEN31350  
GEN31360

```
390          CONTINUE
395          TEMP=TEMP+NPTIP(JJ)
          IF (JJ.EQ.5) TEMP2=TEMP-NPTIP(JJ)
          LNB=LINE
          NLNIP(JJ)=LNB-LNT
              IF (JJ.EQ.1) I2PT21=TEMP+21
              IF (JJ.EQ.1) I2PT41=TEMP+41
              IF (JJ.EQ.1) I2PT46=TEMP+46
              IF (JJ.EQ.1) I2PT51=TEMP+51
              IF (JJ.EQ.4) I5PT21=TEMP+21
              IF (JJ.EQ.4) I5PT41=TEMP+41
              IF (JJ.EQ.4) I5PT46=TEMP+46
              IF (JJ.EQ.4) I5PT51=TEMP+51
400          CONTINUE
          DO 430 JJ=1,2
              LNT=LINE
              START=TEMP1+39
              STOP=TEMP1+40
              LINE=LINE+1
              WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                  DO 405 J=1,11
                      START=START+1
                      STOP=STOP+1
                      LINE=LINE+1
                      WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                          IF ((J.EQ.3).OR.(J.EQ.7)) THEN
                              START=START+1
                              STOP=STOP+1
                          ENDIF
405          CONTINUE
              START=TEMP1+39
              STOP=TEMP1+44
              LINE=LINE+1
              WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                  DO 410 J=1,9
                      START=START+1
                      STOP=STOP+1
                      LINE=LINE+1
                      WRITE(12,250) LINE,COLOR,SOLID,START,STOP
410          CONTINUE
              START=TEMP1+22
              STOP=TEMP1+40
              LINE=LINE+1
              WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                  DO 415 J=1,2
                      START=START-1
```

GEN31370  
GEN31380  
GEN31390  
GEN31400  
GEN31410  
GEN31420  
GEN31430  
GEN31440  
GEN31450  
GEN31460  
GEN31470  
GEN31480  
GEN31490  
GEN31500  
GEN31510  
GEN31520  
GEN31530  
GEN31540  
GEN31550  
GEN31560  
GEN31570  
GEN31580  
GEN31590  
GEN31600  
GEN31610  
GEN31620  
GEN31630  
GEN31640  
GEN31650  
GEN31660  
GEN31670  
GEN31680  
GEN31690  
GEN31700  
GEN31710  
GEN31720  
GEN31730  
GEN31740  
GEN31750  
GEN31760  
GEN31770  
GEN31780  
GEN31790  
GEN31800  
GEN31810  
GEN31820  
GEN31830  
GEN31840  
GEN31850  
GEN31860  
GEN31870  
GEN31880  
GEN31890  
GEN31900  
GEN31910  
GEN31920  
GEN31930  
GEN31940  
GEN31950  
GEN31960  
GEN31970  
GEN31980  
GEN31990  
GEN32000

```

                                STOP=STOP+1
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
415      CONTINUE
                                START=TEMP1+32
                                STOP=TEMP1+43
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                START=TEMP1+33
                                STOP=TEMP1+39
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                IF(JJ.EQ.2) GO TO 425
                                START=TEMP1+39
                                STOP=TEMP1+39+NPTIP(JJ)
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                DO 420 J=1,14
                                    START=START+1
                                    STOP=STOP+1
                                    LINE=LINE+1
                                    WRITE(12,250) LINE,COLOR,SOLID,START,STOP
420      CONTINUE
425      TEMP1=TEMP1+NPTIP(JJ)
                                LNB=LINE
                                NLNIP1(JJ)=LNB-LNT
430      CONTINUE
                                DO 460 JJ=5,IEND
                                    LNT=LINE
                                    START=TEMP2+39
                                    STOP=TEMP2+40
                                    LINE=LINE+1
                                    WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                    DO 435 J=1,11
                                        START=START+1
                                        STOP=STOP+1
                                        LINE=LINE+1
                                        WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                        IF((J.EQ.3).OR.(J.EQ.7)) THEN
                                            START=START+1
                                            STOP=STOP+1
                                        ENDIF
435      CONTINUE
                                START=TEMP2+39
                                STOP=TEMP2+44
```

```

GEN32010
GEN32020
GEN32030
GEN32040
GEN32050
GEN32060
GEN32070
GEN32080
GEN32090
GEN32100
GEN32110
GEN32120
GEN32130
GEN32140
GEN32150
GEN32160
GEN32170
GEN32180
GEN32190
GEN32200
GEN32210
GEN32220
GEN32230
GEN32240
GEN32250
GEN32260
GEN32270
GEN32280
GEN32290
GEN32300
GEN32310
GEN32320
GEN32330
GEN32340
GEN32350
GEN32360
GEN32370
GEN32380
GEN32390
GEN32400
GEN32410
GEN32420
GEN32430
GEN32440
GEN32450
GEN32460
GEN32470
GEN32480
GEN32490
GEN32500
GEN32510
GEN32520
GEN32530
GEN32540
GEN32550
GEN32560
GEN32570
GEN32580
GEN32590
GEN32600
GEN32610
GEN32620
GEN32630
GEN32640
```

```
LINE=LINE+1
WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      DO 440 J=1,9

          START=START+1
          STOP=STOP+1
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

440      CONTINUE

      START=TEMP2+22
      STOP=TEMP2+40
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      DO 445 J=1,2

          START=START-1
          STOP=STOP+1
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

445      CONTINUE

      START=TEMP2+32
      STOP=TEMP2+43
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      START=TEMP2+33
      STOP=TEMP2+39
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      IF(JJ.EQ.IEND) GO TO 455

      START=TEMP2+39
      STOP=TEMP2+39+NPTIP(JJ)
      LINE=LINE+1
      WRITE(12,250) LINE,COLOR,SOLID,START,STOP

      DO 450 J=1,14

          START=START+1
          STOP=STOP+1
          LINE=LINE+1
          WRITE(12,250) LINE,COLOR,SOLID,START,STOP

450      CONTINUE

455      TEMP2=TEMP2+NPTIP(JJ)
          LNB=LINE
          NLNIP2(JJ)=LNB-LNT

460      CONTINUE

      ENDIF

      NLIP=LINE-NLEP-NLRC+6
      WRITE(NO,*) 'NUMBER OF LINES IN INTAKE PORT  = ',NLIP
      NLNTL=LINE
```

```
GEN32650
GEN32660
GEN32670
GEN32680
GEN32690
GEN32700
GEN32710
GEN32720
GEN32730
GEN32740
GEN32750
GEN32760
GEN32770
GEN32780
GEN32790
GEN32800
GEN32810
GEN32820
GEN32830
GEN32840
GEN32850
GEN32860
GEN32870
GEN32880
GEN32890
GEN32900
GEN32910
GEN32920
GEN32930
GEN32940
GEN32950
GEN32960
GEN32970
GEN32980
GEN32990
GEN33000
GEN33010
GEN33020
GEN33030
GEN33040
GEN33050
GEN33060
GEN33070
GEN33080
GEN33090
GEN33100
GEN33110
GEN33120
GEN33130
GEN33140
GEN33150
GEN33160
GEN33170
GEN33180
GEN33190
GEN33200
GEN33210
GEN33220
GEN33230
GEN33240
GEN33250
GEN33260
GEN33270
GEN33280
```

```
C *****
C *
C *          BEGIN LINE GENERATION OF THE SPARK PLUG          *
C *
C *****
```

```
DO 490 JJJ=1,ISP
```

```
DO 485 JJ=1,IEND
```

```
LNT=LINE
```

```
START=NPTTL+1
```

```
STOP=NPTTL+2
```

```
LINE=LINE+1
```

```
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

```
DO 465 J=1,34
```

```
START=START+1
```

```
STOP=STOP+1
```

```
LINE=LINE+1
```

```
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

```
IF (J.EQ.6) STOP=STOP-8
```

```
IF (J.EQ.7) START=START-1
```

```
IF (J.EQ.7) STOP=STOP+7
```

```
IF (J.EQ.15) STOP=STOP-8
```

```
IF (J.EQ.16) STOP=STOP+8
```

```
IF (J.EQ.23) STOP=STOP-8
```

```
IF (J.EQ.24) STOP=STOP+1
```

```
IF (J.EQ.27) STOP=STOP+1
```

```
IF (J.EQ.29) STOP=STOP-8
```

```
IF (J.EQ.30) START=START-5
```

```
IF (J.EQ.30) STOP=STOP+7
```

```
IF (J.EQ.32) START=START+1
```

```
IF (J.EQ.32) STOP=STOP+1
```

465

```
CONTINUE
```

```
START=NPTTL+1
```

```
STOP=NPTTL+10
```

```
LINE=LINE+1
```

```
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

```
DO 470 J=1,14
```

```
START=START+1
```

```
STOP=STOP+1
```

```
LINE=LINE+1
```

```
WRITE(12,250) LINE,COLOR,SOLID,START,STOP
```

```
IF (J.EQ.6) START=START+1
```

470

```
CONTINUE
```

```
GEN33290
GEN33300
GEN33310
GEN33320
GEN33330
GEN33340
GEN33350
GEN33360
GEN33370
GEN33380
GEN33390
GEN33400
GEN33410
GEN33420
GEN33430
GEN33440
GEN33450
GEN33460
GEN33470
GEN33480
GEN33490
GEN33500
GEN33510
GEN33520
GEN33530
GEN33540
GEN33550
GEN33560
GEN33570
GEN33580
GEN33590
GEN33600
GEN33610
GEN33620
GEN33630
GEN33640
GEN33650
GEN33660
GEN33670
GEN33680
GEN33690
GEN33700
GEN33710
GEN33720
GEN33730
GEN33740
GEN33750
GEN33760
GEN33770
GEN33780
GEN33790
GEN33800
GEN33810
GEN33820
GEN33830
GEN33840
GEN33850
GEN33860
GEN33870
GEN33880
GEN33890
GEN33900
GEN33910
GEN33920
```

```
IF (JJ.EQ.IEND) GO TO 480
START=NPTTL+1
STOP=NPTTL+1+NPTSP (JJ)
LINE=LINE+1
WRITE (12,250) LINE,COLOR,SOLID,START,STOP

DO 475 J=1,29

    START=START+1
    STOP=STOP+1
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP

475    CONTINUE

480    NPTTL=NPTTL+NPTSP (JJ)
    LNB=LINE
    NLNSP (JJ)=LNB-LNT

485    CONTINUE

490    CONTINUE

NLSP=LINE-NLRC-NLEP-NLIP+6
WRITE (NO,*) 'NUMBER OF LINES IN SPARK PLUG(S) = ',NLSP

IF (IIPQUE.EQ.1) THEN

    START=I2PT21
    STOP=I2PT41
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN1=LINE

    START=I2PT41
    STOP=I2PT46
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN2=LINE

    START=I2PT46
    STOP=I2PT51
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN3=LINE

    START=I2PT51
    STOP=I5PT51
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN4=LINE

    START=I5PT51
    STOP=I5PT46
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN5=LINE

    START=I5PT46
    STOP=I5PT41
    LINE=LINE+1
    WRITE (12,250) LINE,COLOR,SOLID,START,STOP
    ITCN6=LINE
```

GEN33930  
GEN33940  
GEN33950  
GEN33960  
GEN33970  
GEN33980  
GEN33990  
GEN34000  
GEN34010  
GEN34020  
GEN34030  
GEN34040  
GEN34050  
GEN34060  
GEN34070  
GEN34080  
GEN34090  
GEN34100  
GEN34110  
GEN34120  
GEN34130  
GEN34140  
GEN34150  
GEN34160  
GEN34170  
GEN34180  
GEN34190  
GEN34200  
GEN34210  
GEN34220  
GEN34230  
GEN34240  
GEN34250  
GEN34260  
GEN34270  
GEN34280  
GEN34290  
GEN34300  
GEN34310  
GEN34320  
GEN34330  
GEN34340  
GEN34350  
GEN34360  
GEN34370  
GEN34380  
GEN34390  
GEN34400  
GEN34410  
GEN34420  
GEN34430  
GEN34440  
GEN34450  
GEN34460  
GEN34470  
GEN34480  
GEN34490  
GEN34500  
GEN34510  
GEN34520  
GEN34530  
GEN34540  
GEN34550  
GEN34560



```

                                START=I5PT41
                                STOP=I5PT21
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                ITCN7=LINE

                                START=I5PT21
                                STOP=I2PT21
                                LINE=LINE+1
                                WRITE(12,250) LINE,COLOR,SOLID,START,STOP
                                ITCN8=LINE

ENDIF

WRITE(NO,*) 'TOTAL NUMBER OF LINES = ',LINE
WRITE(NO,*) ' '
WRITE(NO,*) ' '

WRITE(NO,*) 'LINE LABELS OF RC SEGMENTS ARE          1 TO',NLRC
WRITE(NO,*) 'LINE LABELS OF EXHST PORT ARE ',NLRC+1,'TO',NLEP+NLRC
WRITE(NO,*) 'LINE LABELS OF INTKE PORT ARE',NLRC+NLEP+1,'TO',NLIP+
#NLRC+NLEP
WRITE(NO,*) 'LINE LABELS OF SPRK PLUG ARE',NLRC+NLEP+NLIP+1,'TO',
#NLSP+NLRC+NLEP+NLIP
WRITE(NO,*) ' '
WRITE(NO,*) ' '

WRITE(12,245)
WRITE(12,245)

C *****
C *
C *   END GENERATION OF THE LINES IN UNIVERSAL FORMAT
C *
C *****

C *****
C *
C *   BEGIN SPLINE GENERATION OF THE INNER AND OUTER SHELLS,
C *   RIBS, AND STIFFENED CHANNELS IN UNIVERSAL FORMAT
C *   (NOTE THAT THERE ARE NO SPLINES IN ANY OF THE PORTS)
C *
C *****

495  WRITE(12,495)
    FORMAT(4X,'28')
    SPLINE=1
    NUMB=4

    PT1=LASTPT-34
    PT2=1
    PT3=6
    PT4=11
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
520  FORMAT(5I10)
    WRITE(12,525) PT1,PT2,PT3,PT4
525  FORMAT(4I10)

    DO 560 I=1,NUMBER

        IF(RIBTYP(I).EQ.1) THEN

            DO 530 J=1,4

```

```

GEN34570
GEN34580
GEN34590
GEN34600
GEN34610
GEN34620
GEN34630
GEN34640
GEN34650
GEN34660
GEN34670
GEN34680
GEN34690
GEN34700
GEN34710
GEN34720
GEN34730
GEN34740
GEN34750
GEN34760
GEN34770
GEN34780
GEN34790
GEN34800
GEN34810
GEN34820
GEN34830
GEN34840
GEN34850
GEN34860
GEN34870
GEN34880
GEN34890
GEN34900
GEN34910
GEN34920
GEN34930
GEN34940
GEN34950
GEN34960
GEN34970
GEN34980
GEN34990
GEN35000
GEN35010
GEN35020
GEN35030
GEN35040
GEN35050
GEN35060
GEN35070
GEN35080
GEN35090
GEN35100
GEN35110
GEN35120
GEN35130
GEN35140
GEN35150
GEN35160
GEN35170
GEN35180
GEN35190
GEN35200

```

```
PT1=PT1+1
IF ((RIBTYP(I-1).EQ.2).AND.(J.EQ.3)) PT1=PT1+1
PT2=PT2+1
PT3=PT3+1
PT4=PT4+1
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4

530 CONTINUE

PT1=PT1+26
IF(RIBTYP(I-1).EQ.2) PT1=PT1+3
PT2=PT2+26
PT3=PT3+26
PT4=PT4+26
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4

DO 535 J=1,4

    PT1=PT1+1
    IF ((RIBTYP(I-1).EQ.2).AND.(J.EQ.3)) PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

535 CONTINUE

IF(I.EQ.1) THEN

    PT1=11

ELSE

    PT1=PT1+11
    IF(CNLTYP(I-1).EQ.2) PT1=PT1+8

ENDIF

PT2=PT2-19
PT3=PT3-19
PT4=PT4-19
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4

DO 540 J=1,4

    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

540 CONTINUE
```

```
PT1=PT1+26
PT2=PT2+26
PT3=PT3+26
PT4=PT4+26
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4

DO 545 J=1,4

    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

545    CONTINUE

ELSE IF(RIBTYP(I).EQ.2) THEN

DO 547 J=1,2
    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

547    CONTINUE

    PT1=PT1+2
    IF(RIBTYP(I-1).EQ.1) PT1=PT1-1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+2
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

    PT1=PT1+29
    IF(RIBTYP(I-1).EQ.1) PT1=PT1-3
    PT2=PT2+30
    PT3=PT3+30
    PT4=PT4+29
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

DO 549 J=1,2
    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
```

GEN35850  
GEN35860  
GEN35870  
GEN35880  
GEN35890  
GEN35900  
GEN35910  
GEN35920  
GEN35930  
GEN35940  
GEN35950  
GEN35960  
GEN35970  
GEN35980  
GEN35990  
GEN36000  
GEN36010  
GEN36020  
GEN36030  
GEN36040  
GEN36050  
GEN36060  
GEN36070  
GEN36080  
GEN36090  
GEN36100  
GEN36110  
GEN36120  
GEN36130  
GEN36140  
GEN36150  
GEN36160  
GEN36170  
GEN36180  
GEN36190  
GEN36200  
GEN36210  
GEN36220  
GEN36230  
GEN36240  
GEN36250  
GEN36260  
GEN36270  
GEN36280  
GEN36290  
GEN36300  
GEN36310  
GEN36320  
GEN36330  
GEN36340  
GEN36350  
GEN36360  
GEN36370  
GEN36380  
GEN36390  
GEN36400  
GEN36410  
GEN36420  
GEN36430  
GEN36440  
GEN36450  
GEN36460  
GEN36470  
GEN36480

```
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

549 CONTINUE

```
PT1=PT1+2
IF(RIBTYP(I-1).EQ.1) PT1=PT1-1
PT2=PT2+1
PT3=PT3+1
PT4=PT4+2
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

```
PT1=PT1+1
PT2=PT2+1
PT3=PT3+1
PT4=PT4+1
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

```
PT1=PT1+11
IF(CNLTYP(I-1).EQ.2) PT1=PT1+8
PT2=PT2-22
PT3=PT3-21
PT4=PT4-21
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

DO 550 J=1,5

```
PT1=PT1+1
PT2=PT2+1
PT3=PT3+1
PT4=PT4+1
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

550 CONTINUE

```
PT1=PT1+29
PT2=PT2+29
PT3=PT3+29
PT4=PT4+29
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

DO 555 J=1,5

```
PT1=PT1+1
PT2=PT2+1
PT3=PT3+1
PT4=PT4+1
SPLINE=SPLINE+1
WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
WRITE(12,525) PT1,PT2,PT3,PT4
```

555 CONTINUE

GEN36490  
GEN36500  
GEN36510  
GEN36520  
GEN36530  
GEN36540  
GEN36550  
GEN36560  
GEN36570  
GEN36580  
GEN36590  
GEN36600  
GEN36610  
GEN36620  
GEN36630  
GEN36640  
GEN36650  
GEN36660  
GEN36670  
GEN36680  
GEN36690  
GEN36700  
GEN36710  
GEN36720  
GEN36730  
GEN36740  
GEN36750  
GEN36760  
GEN36770  
GEN36780  
GEN36790  
GEN36800  
GEN36810  
GEN36820  
GEN36830  
GEN36840  
GEN36850  
GEN36860  
GEN36870  
GEN36880  
GEN36890  
GEN36900  
GEN36910  
GEN36920  
GEN36930  
GEN36940  
GEN36950  
GEN36960  
GEN36970  
GEN36980  
GEN36990  
GEN37000  
GEN37010  
GEN37020  
GEN37030  
GEN37040  
GEN37050  
GEN37060  
GEN37070  
GEN37080  
GEN37090  
GEN37100  
GEN37110  
GEN37120

```
ENDIF
IF (RIBTYP(I) .EQ. 1) THEN
  IF (CNLTYP(I) .EQ. 2) THEN
    PT1=PT1+16
    PT2=PT2+13
    PT3=PT3+10
    PT4=PT4+7
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

    PT1=PT1+1
    PT2=PT2+1
    PT3=PT3+1
    PT4=PT4+1
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

    IF (I.EQ.NUMBER) GO TO 560

    PT1=PT1-36
    PT2=PT2+5
    PT3=PT3+8
    PT4=PT4+11
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

  ELSE IF (CNLTYP(I) .EQ. 1) THEN
    IF (I.EQ.NUMBER) GO TO 560

    PT1=PT1-19
    PT2=PT2+11
    PT3=PT3+11
    PT4=PT4+11
    SPLINE=SPLINE+1
    WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
    WRITE(12,525) PT1,PT2,PT3,PT4

  ENDIF
ELSE IF (RIBTYP(I) .EQ. 2) THEN
  IF (I.EQ.NUMBER) GO TO 560

  PT1=PT1-21
  PT2=PT2+13
  PT3=PT3+12
  PT4=PT4+11
  SPLINE=SPLINE+1
  WRITE(12,520) SPLINE,SDC,COLOR,SOLID,NUMB
  WRITE(12,525) PT1,PT2,PT3,PT4

ENDIF
560 CONTINUE
NLNTL=NLNTL+SPLINE
```

GEN37130  
GEN37140  
GEN37150  
GEN37160  
GEN37170  
GEN37180  
GEN37190  
GEN37200  
GEN37210  
GEN37220  
GEN37230  
GEN37240  
GEN37250  
GEN37260  
GEN37270  
GEN37280  
GEN37290  
GEN37300  
GEN37310  
GEN37320  
GEN37330  
GEN37340  
GEN37350  
GEN37360  
GEN37370  
GEN37380  
GEN37390  
GEN37400  
GEN37410  
GEN37420  
GEN37430  
GEN37440  
GEN37450  
GEN37460  
GEN37470  
GEN37480  
GEN37490  
GEN37500  
GEN37510  
GEN37520  
GEN37530  
GEN37540  
GEN37550  
GEN37560  
GEN37570  
GEN37580  
GEN37590  
GEN37600  
GEN37610  
GEN37620  
GEN37630  
GEN37640  
GEN37650  
GEN37660  
GEN37670  
GEN37680  
GEN37690  
GEN37700  
GEN37710  
GEN37720  
GEN37730  
GEN37740  
GEN37750  
GEN37760

```

WRITE(NO,*)'NO. IN RC SEGMENT & TOTAL NUMBER OF SPLINES = ',SPLINEGEN37770
WRITE(NO,*) ' ' GEN37780
WRITE(NO,*) ' ' GEN37790

```

```

WRITE(12,245) GEN37800

```

```

C *****
C *
C * END GENERATION OF THE SPLINES IN UNIVERSAL FORMAT *
C *
C ***** GEN37810

```

```

WRITE(NO,565) GEN37820
565 FORMAT('0','THE POINTS, LINES AND SPLINES HAVE BEEN GENERATED.') GEN37830
WRITE(NO,566) GEN37840
566 FORMAT('0','THE PROGRAM CAN BE EXITED AT THIS POINT TO VERIFY') GEN37850
WRITE(NO,567) GEN37860
567 FORMAT('0','THAT THE INTENDED GEOMETRY IS CREATED.') GEN37870
WRITE(NO,570) GEN37880
570 FORMAT('0','ENTER A ...') GEN37890
WRITE(NO,575) GEN37900
575 FORMAT('0',' 1 TO CONTINUE MODEL GENERATION OR ...') GEN37910
WRITE(NO,580) GEN37920
580 FORMAT('0',' 2 TO STOP MODEL GENERATION.') GEN37930

```

```

READ(5,*) ICONTI GEN37940
IF (STATUS.NE.0) WRITE(8,*) ICONTI GEN37950
WRITE(NO,*) ICONTI GEN37960
WRITE(20,*) ICONTI GEN37970

```

```

IF (ICONTI.EQ.2) THEN GEN37980
PRINT*, ' ' GEN37990
PRINT*, ' ' GEN38000
PRINT*, ' THE UNIVERSAL HAS BEEN CREATED.' GEN38010
PRINT*, ' ' GEN38020
PRINT*, ' BEFORE THE UNIVERSAL FILE CAN BE READ,' GEN38030
PRINT*, ' THE USER MUST ENTER THE SDRC SOFTWARE.' GEN38040

```

```

PRINT*, ' ' GEN38050
PRINT*, ' NOTE THAT A FILE CALLED "MODEL DATA" CAN NOT' GEN38060
PRINT*, ' EXIST ON YOUR DISK. IF IT DOES, AN ERROR WILL' GEN38070
PRINT*, ' RESULT WHEN YOU ENTER THE SDRC SOFTWARE. TO' GEN38080
PRINT*, ' CORRECT THE ERROR, CHANGE THE NAME OF THE' GEN38090
PRINT*, ' "MODEL DATA" FILE THAT EXISTS ON YOUR DISK.' GEN38100
PRINT*, ' (BEFORE YOU ENTER THE SDRC SOFTWARE)' GEN38110
PRINT*, ' ' GEN38120
PRINT*, ' AFTER ENTERING THE SDRC SOFTWARE IN THE PROGRAM MODE,' GEN38130
PRINT*, ' RESPOND "R" (FOR RUN) TO THE FIRST QUESTION,' GEN38140
PRINT*, ' RESPOND "GO" TO THE SECOND QUESTION AND' GEN38150
PRINT*, ' IN RESPONSE TO THE THIRD QUESTION,' GEN38160
PRINT*, ' ENTER THE TERMINAL TYPE THAT YOU ARE USING.' GEN38170

```

```

CALL GO (ICONTI) GEN38180
STOP GEN38190
ENDIF GEN38200

```

```

WRITE(12,245) GEN38210

```

```

C *****
C *
C * BEGIN GENERATION OF THE EDGES IN UNIVERSAL FORMAT *
C *
C ***** GEN38220

```

```
C *****
      WRITE(12,585)
585  FORMAT(4X,'29')

      SPENTC=3
      LNENTC=1
      NUMB=1
      ISPENL=1
      ILNENL=1
      EDGE=1

      DO 625 III=1,NUMBER
        DO 610 II=1,2
          IF((RIBTYP(III).EQ.2).AND.(II.EQ.2)) THEN
            NMSPS=12
          ELSE
            NMSPS=10
          ENDIF

          DO 600 I=1,NMSPS
            WRITE(12,590) EDGE,COLOR,SOLID,NUMB
590    FORMAT(4I10)
            WRITE(12,595) SPENTC,ISPENL
595    FORMAT(6I10)

            EDGE=EDGE+1
            ISPENL=ISPENL+1

600          CONTINUE

            IF(RIBTYP(III).EQ.2) THEN
              NMLNS=14
            ELSE IF(RIBTYP(III).EQ.1) THEN
              NMLNS=13
            ENDIF

            DO 605 I=1,NMLNS
              WRITE(12,590) EDGE,COLOR,SOLID,NUMB
              WRITE(12,595) LNENTC,ILNENL

              EDGE=EDGE+1
              ILNENL=ILNENL+1

605          CONTINUE
610          CONTINUE

            IF(CNLTYP(III).EQ.2) THEN

              DO 615 J=1,2

                WRITE(12,590) EDGE,COLOR,SOLID,NUMB
                WRITE(12,595) SPENTC,ISPENL
                EDGE=EDGE+1
```

```
GEN38410
GEN38420
GEN38430
GEN38440
GEN38450
GEN38460
GEN38470
GEN38480
GEN38490
GEN38500
GEN38510
GEN38520
GEN38530
GEN38540
GEN38550
GEN38560
GEN38570
GEN38580
GEN38590
GEN38600
GEN38610
GEN38620
GEN38630
GEN38640
GEN38650
GEN38660
GEN38670
GEN38680
GEN38690
GEN38700
GEN38710
GEN38720
GEN38730
GEN38740
GEN38750
GEN38760
GEN38770
GEN38780
GEN38790
GEN38800
GEN38810
GEN38820
GEN38830
GEN38840
GEN38850
GEN38860
GEN38870
GEN38880
GEN38890
GEN38900
GEN38910
GEN38920
GEN38930
GEN38940
GEN38950
GEN38960
GEN38970
GEN38980
GEN38990
GEN39000
GEN39010
GEN39020
GEN39030
GEN39040
```

```

        ISPENL=ISPENL+1
615      CONTINUE
        DO 620 J=1,2
            WRITE(12,590) EDGE,COLOR,SOLID,NUMB
            WRITE(12,595) LNENTC,ILNENL
            EDGE=EDGE+1
            ILNENL=ILNENL+1
620      CONTINUE
        ENDIF
625      CONTINUE
        TEMPED=EDGE-1
        IF (IEPQUE.EQ.1) THEN
            DO 630 J=EDGE,LINE+SPLINE-6
                WRITE(12,590) EDGE,COLOR,SOLID,NUMB
                WRITE(12,595) LNENTC,ILNENL
                ILNENL=ILNENL+1
                EDGE=EDGE+1
            IF ((EDGE.EQ.LINE+SPLINE-5).OR.(EDGE.EQ.LINE+SPLINE-7)) THEN
                NUMB=3
                WRITE(12,590) EDGE,COLOR,SOLID,NUMB
                WRITE(12,595) LNENTC,ILNENL,LNENTC,ILNENL+1,LNENTC,ILNENL+2
                ILNENL=ILNENL+3
                EDGE=EDGE+1
                NUMB=1
            ENDIF
630      CONTINUE
        ENDIF
        WRITE(12,245)
        WRITE(12,245)
        NERC=EDGE-1-NLEP-NLIP-NLSP
        NEEP=EDGE-1-SPLINE-NLRC-NLIP-NLSP
        NEIP=EDGE-1-SPLINE-NLRC-NLEP-NLSP
        NESP=EDGE-1-SPLINE-NLRC-NLEP-NLIP+6
        WRITE(NO,*) ' '
        WRITE(NO,*) ' '
        WRITE(NO,*) 'NUMBER OF EDGES IN RC SEGMENTS = ',NERC
        WRITE(NO,*) 'NUMBER OF EDGES IN EXHAUST PORT = ',NEEP
        WRITE(NO,*) 'NUMBER OF EDGES IN INTAKE PORT = ',NEIP
        WRITE(NO,*) 'NUMBER OF EDGES IN SPARK PLUG(S) = ',NESP
        WRITE(NO,*) 'TOTAL NUMBER OF EDGES = ',EDGE-1
        WRITE(NO,*) ' '
        WRITE(NO,*) ' '
        WRITE(NO,*) 'EDGE LABELS OF RC SEGMENTS ARE 1 TO',NERC
        WRITE(NO,*) 'EDGE LABELS OF EXHST PORT ARE ',NERC+1,'TO',NEEP+NERC
        WRITE(NO,*) 'EDGE LABELS OF INTKE PORT ARE ',NERC+NEEP+1,'TO',NEIP+

```

GEN39050  
 GEN39060  
 GEN39070  
 GEN39080  
 GEN39090  
 GEN39100  
 GEN39110  
 GEN39120  
 GEN39130  
 GEN39140  
 GEN39150  
 GEN39160  
 GEN39170  
 GEN39180  
 GEN39190  
 GEN39200  
 GEN39210  
 GEN39220  
 GEN39230  
 GEN39240  
 GEN39250  
 GEN39260  
 GEN39270  
 GEN39280  
 GEN39290  
 GEN39300  
 GEN39310  
 GEN39320  
 GEN39330  
 GEN39340  
 GEN39350  
 GEN39360  
 GEN39370  
 GEN39380  
 GEN39390  
 GEN39400  
 GEN39410  
 GEN39420  
 GEN39430  
 GEN39440  
 GEN39450  
 GEN39460  
 GEN39470  
 GEN39480  
 GEN39490  
 GEN39500  
 GEN39510  
 GEN39520  
 GEN39530  
 GEN39540  
 GEN39550  
 GEN39560  
 GEN39570  
 GEN39580  
 GEN39590  
 GEN39600  
 GEN39610  
 GEN39620  
 GEN39630  
 GEN39640  
 GEN39650  
 GEN39660  
 GEN39670  
 GEN39680



```
#NERC+NEEP
WRITE(NO,*)'EDGE LABELS OF SPRK PLUGS ARE ',NERC+NEEP+NEIP+1,'TO',
#NESP+NERC+NEEP+NEIP
WRITE(NO,*) ' '
WRITE(NO,*) ' '
C *****
C *
C *      END GENERATION OF THE EDGES IN UNIVERSAL FORMAT
C *
C *****
C *****
C *
C *      BEGIN GENERATION OF THE SURFACES OF THE INNER AND
C *      OUTER SHELLS, AND RIBS IN UNIVERSAL FORMAT
C *
C *****
        ISRCT=0
        WRITE(12,635)
635   FORMAT(4X,'30')
        EDGE=TEMPED
        SURFAC=1
        DASH=2
        NUMB=4
        GRADE=2
        ED1=1
        IF(RIBTYP(NUMBER).EQ.2) THEN
            EDGE=EDGE-2
        ENDIF
        ED2=EDGE-8
        ED3=6
        ED4=15
        WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
640   FORMAT(5I10)
        WRITE(12,645) ED1,ED2,ED3,ED4
645   FORMAT(4I10)
        DO 690 I=1,NUMBER
            IF(RIBTYP(I).EQ.1) THEN
                DO 675 JJ=1,2
                    DO 650 J=1,4
                        ED1=ED1+1
                        ED2=ED2+1
                        IF (((J.EQ.3).AND.(JJ.EQ.1)).AND.(RIBTYP(I-1).EQ.2))
#                            ED2=ED2+1
                        ED3=ED3+1
                        ED4=ED4+1
                        SURFAC=SURFAC+1
                        WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
                        WRITE(12,645) ED1,ED2,ED3,ED4
650   CONTINUE
```

```
GEN39690
GEN39700
GEN39710
GEN39720
GEN39730
GEN39740
GEN39750
GEN39760
GEN39770
GEN39780
GEN39790
GEN39800
GEN39810
GEN39820
GEN39830
GEN39840
GEN39850
GEN39860
GEN39870
GEN39880
GEN39890
GEN39900
GEN39910
GEN39920
GEN39930
GEN39940
GEN39950
GEN39960
GEN39970
GEN39980
GEN39990
GEN40000
GEN40010
GEN40020
GEN40030
GEN40040
GEN40050
GEN40060
GEN40070
GEN40080
GEN40090
GEN40100
GEN40110
GEN40120
GEN40130
GEN40140
GEN40150
GEN40160
GEN40170
GEN40180
GEN40190
GEN40200
GEN40210
GEN40220
GEN40230
GEN40240
GEN40250
GEN40260
GEN40270
GEN40280
GEN40290
GEN40300
GEN40310
GEN40320
```

```
ED1=ED1-4
ED2=ED2-8
  IF ((RIBTYP(I-1).EQ.2).AND.(JJ.EQ.1)) ED2=ED2-1
ED3=ED3-8
ED4=ED4-8
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

IF(JJ.EQ.1) THEN

  ED1=ED1+1
  ED2=ED2+1
  ED3=ED3+1
  ED4=ED4+1
  SURFAC=SURFAC+1
  WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
  WRITE(12,645) ED1,ED2,ED3,ED4

  ED1=ED1+2
  ED2=ED2+2
  ED3=ED3+2
  ED4=ED4+2
  SURFAC=SURFAC+1
  WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
  WRITE(12,645) ED1,ED2,ED3,ED4

  ED1=ED1+2
  ED2=ED2+6
  IF(RIBTYP(I-1).EQ.2) ED2=ED2+1
  ED3=ED3+2
  ED4=ED4+6
  SURFAC=SURFAC+1
  WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
  WRITE(12,645) ED1,ED2,ED3,ED4

  ED1=ED1+1
  ED2=ED2+1
  ED3=ED3+1
  ED4=ED4+1
  SURFAC=SURFAC+1
  WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
  WRITE(12,645) ED1,ED2,ED3,ED4

  ED1=ED1+2
  ED2=ED2+2
  ED3=ED3+2
  ED4=ED4+2
  SURFAC=SURFAC+1
  WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
  WRITE(12,645) ED1,ED2,ED3,ED4

ELSE IF(JJ.EQ.2) THEN

  DO 655 JJJ=1,3

    ED1=ED1+1
    ED2=ED2+1
    ED3=ED3+1
    ED4=ED4+1
    SURFAC=SURFAC+1
    WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
    WRITE(12,645) ED1,ED2,ED3,ED4
```

```
655          CONTINUE                                GEN40970
                                                    GEN40980
                                                    GEN40990
                                                    GEN41000
                                                    GEN41010
                                                    GEN41020
                                                    GEN41030
                                                    GEN41040
                                                    GEN41050
                                                    GEN41060
                                                    GEN41070
                                                    GEN41080
                                                    GEN41090
                                                    GEN41100
                                                    GEN41110
                                                    GEN41120
                                                    GEN41130
                                                    GEN41140
                                                    GEN41150
                                                    GEN41160
                                                    GEN41170
                                                    GEN41180
                                                    GEN41190
                                                    GEN41200
                                                    GEN41210
                                                    GEN41220
                                                    GEN41230
                                                    GEN41240
                                                    GEN41250
                                                    GEN41260
                                                    GEN41270
                                                    GEN41280
                                                    GEN41290
                                                    GEN41300
                                                    GEN41310
                                                    GEN41320
                                                    GEN41330
                                                    GEN41340
                                                    GEN41350
                                                    GEN41360
                                                    GEN41370
                                                    GEN41380
                                                    GEN41390
                                                    GEN41400
                                                    GEN41410
                                                    GEN41420
                                                    GEN41430
                                                    GEN41440
                                                    GEN41450
                                                    GEN41460
                                                    GEN41470
                                                    GEN41480
                                                    GEN41490
                                                    GEN41500
                                                    GEN41510
                                                    GEN41520
                                                    GEN41530
                                                    GEN41540
                                                    GEN41550
                                                    GEN41560
                                                    GEN41570
                                                    GEN41580
                                                    GEN41590
                                                    GEN41600

          ED1=ED1+2
          ED2=ED2+6
          ED3=ED3+2
          ED4=ED4+6
          SURFAC=SURFAC+1
          WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
          WRITE (12, 645) ED1, ED2, ED3, ED4

          DO 660 JJJ=1, 3

          ED1=ED1+1
          ED2=ED2+1
          ED3=ED3+1
          ED4=ED4+1
          SURFAC=SURFAC+1
          WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
          WRITE (12, 645) ED1, ED2, ED3, ED4

660          CONTINUE

        ENDIF
        ED1=ED1+6
        IF (I.GT.1) GO TO 665

        IF ((I.EQ.1).AND.(JJ.EQ.1)) THEN
          ED2=ED2+20-EDGE
        ELSE IF (JJ.EQ.2) THEN
          ED2=ED2+20
        ENDIF

665        IF ((CNLTYP(I-1).EQ.2).AND.(JJ.EQ.2)) THEN
          ED2=ED2+20
        ELSE IF ((CNLTYP(I-1).EQ.2).AND.(JJ.EQ.1)) THEN
          ED2=ED2+24
        ELSE IF (CNLTYP(I-1).EQ.1) THEN
          ED2=ED2+20
        ENDIF

        ED3=ED3+6
        ED4=ED4-12
        SURFAC=SURFAC+1
        WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
        WRITE (12, 645) ED1, ED2, ED3, ED4

        DO 670 J=1, 3

        ED1=ED1+1
        ED2=ED2+1
        ED3=ED3+1
        ED4=ED4+1
        SURFAC=SURFAC+1
        WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
        WRITE (12, 645) ED1, ED2, ED3, ED4

670      CONTINUE

      IF (JJ.EQ.1) THEN
        ED1=ED1+6
        ED2=ED2-8
        ED3=ED3+10
        ED4=ED4+24
        SURFAC=SURFAC+1
```

```
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ENDIF
```

```
675
```

```
CONTINUE
```

```
ELSE IF(RIBTYP(I).EQ.2) THEN
```

```
DO 680 J=1,2
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
680
```

```
CONTINUE
```

```
ED1=ED1+1
ED2=ED2+2
IF(RIBTYP(I-1).EQ.1) ED2=ED2-1
ED3=ED3+1
ED4=ED4+2
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1-4
ED2=ED2-9
IF(RIBTYP(I-1).EQ.1) ED2=ED2+1
ED3=ED3-8
ED4=ED4-9
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1+2
ED2=ED2+2
ED3=ED3+2
ED4=ED4+2
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1+2
```

```
GEN41610
GEN41620
GEN41630
GEN41640
GEN41650
GEN41660
GEN41670
GEN41680
GEN41690
GEN41700
GEN41710
GEN41720
GEN41730
GEN41740
GEN41750
GEN41760
GEN41770
GEN41780
GEN41790
GEN41800
GEN41810
GEN41820
GEN41830
GEN41840
GEN41850
GEN41860
GEN41870
GEN41880
GEN41890
GEN41900
GEN41910
GEN41920
GEN41930
GEN41940
GEN41950
GEN41960
GEN41970
GEN41980
GEN41990
GEN42000
GEN42010
GEN42020
GEN42030
GEN42040
GEN42050
GEN42060
GEN42070
GEN42080
GEN42090
GEN42100
GEN42110
GEN42120
GEN42130
GEN42140
GEN42150
GEN42160
GEN42170
GEN42180
GEN42190
GEN42200
GEN42210
GEN42220
GEN42230
GEN42240
```

ED2=ED2+7	GEN42250
IF (RIBTYP (I-1) .EQ.1) ED2=ED2-1	GEN42260
ED3=ED3+2	GEN42270
ED4=ED4+7	GEN42280
SURFAC=SURFAC+1	GEN42290
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42300
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42310
	GEN42320
ED1=ED1+1	GEN42330
ED2=ED2+1	GEN42340
ED3=ED3+1	GEN42350
ED4=ED4+1	GEN42360
SURFAC=SURFAC+1	GEN42370
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42380
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42390
	GEN42400
ED1=ED1+2	GEN42410
ED2=ED2+2	GEN42420
ED3=ED3+2	GEN42430
ED4=ED4+2	GEN42440
SURFAC=SURFAC+1	GEN42450
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42460
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42470
	GEN42480
ED1=ED1+6	GEN42490
ED2=ED2+23	GEN42500
IF (RIBTYP (I-1) .EQ.1) ED2=ED2-2	GEN42510
IF (RIBTYP (I-1) .EQ.2) ED2=ED2-2	GEN42520
IF (CNLTYP (I-1) .EQ.2) ED2=ED2+4	GEN42530
ED3=ED3+6	GEN42540
ED4=ED4-13	GEN42550
SURFAC=SURFAC+1	GEN42560
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42570
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42580
	GEN42590
ED1=ED1+1	GEN42600
ED2=ED2+1	GEN42610
ED3=ED3+1	GEN42620
ED4=ED4+1	GEN42630
SURFAC=SURFAC+1	GEN42640
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42650
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42660
	GEN42670
ED1=ED1+2	GEN42680
ED2=ED2+1	GEN42690
ED3=ED3+2	GEN42700
ED4=ED4+1	GEN42710
SURFAC=SURFAC+1	GEN42720
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42730
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42740
	GEN42750
ED1=ED1+1	GEN42760
ED2=ED2+1	GEN42770
ED3=ED3+1	GEN42780
ED4=ED4+1	GEN42790
SURFAC=SURFAC+1	GEN42800
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB	GEN42810
WRITE (12, 645) ED1, ED2, ED3, ED4	GEN42820
	GEN42830
ED1=ED1+6	GEN42840
ED2=ED2-9	GEN42850
ED3=ED3+11	GEN42860
ED4=ED4+27	GEN42870
SURFAC=SURFAC+1	GEN42880

```
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

DO 685 J=1,5

    ED1=ED1+1
    ED2=ED2+1
    ED3=ED3+1
    ED4=ED4+1
    SURFAC=SURFAC+1
    WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
    WRITE(12,645) ED1,ED2,ED3,ED4
```

685

CONTINUE

```
ED1=ED1-5
ED2=ED2-9
ED3=ED3-10
ED4=ED4-9
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+2
ED2=ED2+1
ED3=ED3+2
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+2
ED2=ED2+7
ED3=ED3+2
ED4=ED4+7
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1+2
```

GEN42890  
GEN42900  
GEN42910  
GEN42920  
GEN42930  
GEN42940  
GEN42950  
GEN42960  
GEN42970  
GEN42980  
GEN42990  
GEN43000  
GEN43010  
GEN43020  
GEN43030  
GEN43040  
GEN43050  
GEN43060  
GEN43070  
GEN43080  
GEN43090  
GEN43100  
GEN43110  
GEN43120  
GEN43130  
GEN43140  
GEN43150  
GEN43160  
GEN43170  
GEN43180  
GEN43190  
GEN43200  
GEN43210  
GEN43220  
GEN43230  
GEN43240  
GEN43250  
GEN43260  
GEN43270  
GEN43280  
GEN43290  
GEN43300  
GEN43310  
GEN43320  
GEN43330  
GEN43340  
GEN43350  
GEN43360  
GEN43370  
GEN43380  
GEN43390  
GEN43400  
GEN43410  
GEN43420  
GEN43430  
GEN43440  
GEN43450  
GEN43460  
GEN43470  
GEN43480  
GEN43490  
GEN43500  
GEN43510  
GEN43520

```
ED2=ED2+1
ED3=ED3+2
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ED1=ED1+6
ED2=ED2+23
ED3=ED3+6
ED4=ED4-13
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ED1=ED1+2
ED2=ED2+1
ED3=ED3+2
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4

ENDIF

IF (RIBTYP (I) .EQ. 1) THEN

    IF (CNLTYP (I) .EQ. 1) THEN

        IF (I.EQ.NUMBER) GO TO 690
        ED1=ED1+6
        ED2=ED2-8
        ED3=ED3+10
        ED4=ED4+24
        SURFAC=SURFAC+1
        WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
        WRITE (12, 645) ED1, ED2, ED3, ED4

    ELSE IF (CNLTYP (I) .EQ. 2) THEN
```

```
GEN43530
GEN43540
GEN43550
GEN43560
GEN43570
GEN43580
GEN43590
GEN43600
GEN43610
GEN43620
GEN43630
GEN43640
GEN43650
GEN43660
GEN43670
GEN43680
GEN43690
GEN43700
GEN43710
GEN43720
GEN43730
GEN43740
GEN43750
GEN43760
GEN43770
GEN43780
GEN43790
GEN43800
GEN43810
GEN43820
GEN43830
GEN43840
GEN43850
GEN43860
GEN43870
GEN43880
GEN43890
GEN43900
GEN43910
GEN43920
GEN43930
GEN43940
GEN43950
GEN43960
GEN43970
GEN43980
GEN43990
GEN44000
GEN44010
GEN44020
GEN44030
GEN44040
GEN44050
GEN44060
GEN44070
GEN44080
GEN44090
GEN44100
GEN44110
GEN44120
GEN44130
GEN44140
GEN44150
GEN44160
```

GEN44170  
GEN44180  
GEN44190  
GEN44200  
GEN44210  
GEN44220  
GEN44230  
GEN44240  
GEN44250  
GEN44260  
GEN44270  
GEN44280  
GEN44290  
GEN44300  
GEN44310  
GEN44320  
GEN44330  
GEN44340  
GEN44350  
GEN44360  
GEN44370  
GEN44380  
GEN44390  
GEN44400  
GEN44410  
GEN44420  
GEN44430  
GEN44440  
GEN44450  
GEN44460  
GEN44470  
GEN44480  
GEN44490  
GEN44500  
GEN44510  
GEN44520  
GEN44530  
GEN44540  
GEN44550  
GEN44560  
GEN44570  
GEN44580  
GEN44590  
GEN44600  
GEN44610  
GEN44620  
GEN44630  
GEN44640  
GEN44650  
GEN44660  
GEN44670  
GEN44680  
GEN44690  
GEN44700  
GEN44710  
GEN44720  
GEN44730  
GEN44740  
GEN44750  
GEN44760  
GEN44770  
GEN44780  
GEN44790  
GEN44800

ED1=ED1+6  
ED2=ED2+3  
ED3=ED3+6  
ED4=ED4+13  
SURFAC=SURFAC+1

ISRCT=ISRCT+1  
WRITE(20,\*) SURFAC,I

WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB  
WRITE(12,645) ED1,ED2,ED3,ED4

IF(I.EQ.NUMBER) GO TO 690

ED1=ED1+4  
ED2=ED2-11  
ED3=ED3+8  
ED4=ED4+15  
SURFAC=SURFAC+1  
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB  
WRITE(12,645) ED1,ED2,ED3,ED4

ENDIF

ELSE IF(RIBTYP(I).EQ.2) THEN

IF(I.EQ.NUMBER) GO TO 690  
ED1=ED1+6  
ED2=ED2-9  
ED3=ED3+10  
ED4=ED4+25  
SURFAC=SURFAC+1  
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB  
WRITE(12,645) ED1,ED2,ED3,ED4

ENDIF

690 CONTINUE

TEMPSU=SURFAC  
TEMSU=SURFAC  
EDGE=TEMPED

NSRC=SURFAC  
WRITE(NO,\*) 'NUMBER OF SURFACES IN RC SEGMENTS = ',NSRC

C \*\*\*\*\*  
C \*  
C \* BEGIN GENERATION OF THE SURFACES OF THE  
C \* EXHAUST PORT  
C \*  
C \*\*\*\*\*

IF (IEPQUE.EQ.1) THEN

DO 715 JJ=1,IEND

SURT=SURFAC  
ED1=EDGE+1  
ED2=EDGE+26  
ED3=EDGE+15  
ED4=EDGE+27  
SURFAC=SURFAC+1



```
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
DO 695 J=1,19
```

```
    ED1=ED1+1
    ED2=ED2+1
    ED3=ED3+1
    ED4=ED4+1
    SURFAC=SURFAC+1
    WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
    WRITE(12,645) ED1,ED2,ED3,ED4
```

```
    IF (J.EQ.9) ED4=ED4-24
```

```
    IF (J.EQ.10) THEN
```

```
        ED2=ED2-24
        ED3=ED3-12
        ED4=ED4+12
```

```
    ENDIF
```

```
    IF (J.EQ.11) THEN
```

```
        ED1=ED1+3
        ED2=ED2+23
        ED3=ED3+32
        ED4=ED4+11
```

```
    ENDIF
```

```
    IF (((JJ.EQ.3).OR.(JJ.EQ.4)).AND.(J.EQ.11)) GO TO 700
```

```
    IF (J.EQ.15) THEN
```

```
        ED1=ED1+2
        ED2=ED2+1
        ED4=ED4+1
```

```
    ENDIF
```

```
695      CONTINUE
```

```
700  IF (JJ.EQ.IEND) GO TO 710
```

```
    ED1=EDGE+1
    ED2=EDGE+55
    ED3=EDGE+1+NLNEP(JJ)
    ED4=EDGE+56
```

```
    IF ((JJ.EQ.3).OR.(JJ.EQ.4)) THEN
```

```
        ED2=ED2-18
        ED4=ED4-18
```

```
    ENDIF
```

```
    SURFAC=SURFAC+1
    WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
    WRITE(12,645) ED1,ED2,ED3,ED4
```

```
    DO 705 J=1,53
```

```
GEN44810
GEN44820
GEN44830
GEN44840
GEN44850
GEN44860
GEN44870
GEN44880
GEN44890
GEN44900
GEN44910
GEN44920
GEN44930
GEN44940
GEN44950
GEN44960
GEN44970
GEN44980
GEN44990
GEN45000
GEN45010
GEN45020
GEN45030
GEN45040
GEN45050
GEN45060
GEN45070
GEN45080
GEN45090
GEN45100
GEN45110
GEN45120
GEN45130
GEN45140
GEN45150
GEN45160
GEN45170
GEN45180
GEN45190
GEN45200
GEN45210
GEN45220
GEN45230
GEN45240
GEN45250
GEN45260
GEN45270
GEN45280
GEN45290
GEN45300
GEN45310
GEN45320
GEN45330
GEN45340
GEN45350
GEN45360
GEN45370
GEN45380
GEN45390
GEN45400
GEN45410
GEN45420
GEN45430
GEN45440
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.10) ED4=ED4-12
      IF (J.EQ.11) ED4=ED4+10
      IF (J.EQ.12) ED2=ED2-1
      IF (J.EQ.12) ED4=ED4+1
      IF (J.EQ.23) ED4=ED4-12
      IF (J.EQ.24) ED2=ED2-24
      IF (((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)).AND.
#      (J.EQ.35)) GO TO 710
      IF (J.EQ.35) ED2=ED2+3
      IF (J.EQ.40) ED2=ED2+1
      IF (J.EQ.44) ED2=ED2-12
      IF (J.EQ.45) ED2=ED2+11
      IF (J.EQ.45) ED4=ED4-9
      IF (J.EQ.49) ED2=ED2+1
      IF (J.EQ.49) ED4=ED4+1
705      CONTINUE
710      EDGE=EDGE+NLNEP(JJ)
      SURB=SURFAC
      NSREP(JJ)=SURB-SURT
715      CONTINUE
      ITOTAL=0
      DO 720 J=1,IEND
720      ITOTAL=ITOTAL+NLNEP(J)
      NEDEP=ITOTAL+TEMPED
      TEMSU1=SURFAC
      DO 755 JJ=1,2
      ED1=NEDEP+1
      ED2=NEDEP+13
      ED3=NEDEP+5
      ED4=NEDEP+14
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      DO 725 J=1,7
      ED1=ED1+1
      ED2=ED2+1
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
```

GEN45450  
GEN45460  
GEN45470  
GEN45480  
GEN45490  
GEN45500  
GEN45510  
GEN45520  
GEN45530  
GEN45540  
GEN45550  
GEN45560  
GEN45570  
GEN45580  
GEN45590  
GEN45600  
GEN45610  
GEN45620  
GEN45630  
GEN45640  
GEN45650  
GEN45660  
GEN45670  
GEN45680  
GEN45690  
GEN45700  
GEN45710  
GEN45720  
GEN45730  
GEN45740  
GEN45750  
GEN45760  
GEN45770  
GEN45780  
GEN45790  
GEN45800  
GEN45810  
GEN45820  
GEN45830  
GEN45840  
GEN45850  
GEN45860  
GEN45870  
GEN45880  
GEN45890  
GEN45900  
GEN45910  
GEN45920  
GEN45930  
GEN45940  
GEN45950  
GEN45960  
GEN45970  
GEN45980  
GEN45990  
GEN46000  
GEN46010  
GEN46020  
GEN46030  
GEN46040  
GEN46050  
GEN46060  
GEN46070  
GEN46080

```

      WRITE(12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.3) THEN
        ED2=ED2+1
        ED4=ED4+1
      ENDIF
725      CONTINUE
      ED1=TEMPED+46
      ED2=NEDEP+27
      ED3=NEDEP+1
      ED4=NEDEP+23
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      ED1=ED1-32
      ED2=ED2-4
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      DO 730 J=1,2
        ED1=ED1+1
        ED2=ED2+1
        ED3=ED3+1
        ED4=ED4+1
        SURFAC=SURFAC+1
        WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
        WRITE(12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.1) ED1=ED1+21
730      CONTINUE
      IF (JJ.EQ.2) GO TO 750
      ED1=NEDEP+1
      ED2=NEDEP+28
      ED3=NEDEP+1+NLNEP1(JJ)
      ED4=NEDEP+29
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      DO 735 J=1,11
        ED1=ED1+1
        ED2=ED2+1
        ED3=ED3+1
        ED4=ED4+1
        SURFAC=SURFAC+1
        WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
        WRITE(12,645) ED1,ED2,ED3,ED4
      IF ((J.EQ.3).OR.(J.EQ.7)) THEN
        ED2=ED2+1
```

GEN46090  
GEN46100  
GEN46110  
GEN46120  
GEN46130  
GEN46140  
GEN46150  
GEN46160  
GEN46170  
GEN46180  
GEN46190  
GEN46200  
GEN46210  
GEN46220  
GEN46230  
GEN46240  
GEN46250  
GEN46260  
GEN46270  
GEN46280  
GEN46290  
GEN46300  
GEN46310  
GEN46320  
GEN46330  
GEN46340  
GEN46350  
GEN46360  
GEN46370  
GEN46380  
GEN46390  
GEN46400  
GEN46410  
GEN46420  
GEN46430  
GEN46440  
GEN46450  
GEN46460  
GEN46470  
GEN46480  
GEN46490  
GEN46500  
GEN46510  
GEN46520  
GEN46530  
GEN46540  
GEN46550  
GEN46560  
GEN46570  
GEN46580  
GEN46590  
GEN46600  
GEN46610  
GEN46620  
GEN46630  
GEN46640  
GEN46650  
GEN46660  
GEN46670  
GEN46680  
GEN46690  
GEN46700  
GEN46710  
GEN46720

```

                                ED4=ED4+1                                GEN46730
                                ENDIF                                GEN46740
                                GEN46750
                                GEN46760
735      CONTINUE                                GEN46770
                                GEN46780
                                ED1=NEDEP+13                        GEN46790
                                ED2=NEDEP+28                        GEN46800
                                ED3=NEDEP+13+NLNEP1(JJ)            GEN46810
                                ED4=NEDEP+33                        GEN46820
                                SURFAC=SURFAC+1                    GEN46830
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN46840
                                WRITE(12,645) ED1,ED2,ED3,ED4      GEN46850
                                GEN46860
                                DO 740 J=1,9                        GEN46870
                                ED1=ED1+1                            GEN46880
                                ED2=ED2+1                            GEN46890
                                ED3=ED3+1                            GEN46900
                                ED4=ED4+1                            GEN46910
                                SURFAC=SURFAC+1                    GEN46920
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN46930
                                WRITE(12,645) ED1,ED2,ED3,ED4      GEN46940
                                GEN46950
                                GEN46960
740      CONTINUE                                GEN46970
                                GEN46980
                                ED1=NEDEP+27                        GEN46990
                                ED2=TEMPED+88                        GEN47000
                                ED3=NEDEP+27+NLNEP1(JJ)            GEN47010
                                ED4=NEDEP+28                        GEN47020
                                SURFAC=SURFAC+1                    GEN47030
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN47040
                                WRITE(12,645) ED1,ED2,ED3,ED4      GEN47050
                                GEN47060
                                ED1=ED1-4                            GEN47070
                                ED2=ED2-21                            GEN47080
                                ED3=ED3-4                            GEN47090
                                ED4=ED4+1                            GEN47100
                                SURFAC=SURFAC+1                    GEN47110
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN47120
                                WRITE(12,645) ED1,ED2,ED3,ED4      GEN47130
                                GEN47140
                                DO 745 J=1,3                        GEN47150
                                ED1=ED1+1                            GEN47160
                                ED2=ED2+1                            GEN47170
                                ED3=ED3+1                            GEN47180
                                ED4=ED4+1                            GEN47190
                                SURFAC=SURFAC+1                    GEN47200
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN47210
                                WRITE(12,645) ED1,ED2,ED3,ED4      GEN47220
                                GEN47230
                                IF (J.EQ.2) ED2=ED2+9              GEN47240
                                GEN47250
                                GEN47260
745      CONTINUE                                GEN47270
                                GEN47280
750      NEDEP=NEDEP+NLNEP1(JJ)                        GEN47290
                                TEMPED=TEMPED+NLNEP1(JJ)          GEN47300
                                GEN47310
755      CONTINUE                                GEN47320
                                GEN47330
                                ENDIF                                GEN47340
                                GEN47350
                                NEDEP=NLNEP+NLRC+SPLINE            GEN47360
```

```
TEMPID=NEDEP
TEMP11=NEDEP+NLNIP (1) +NLNIP (2) +NLNIP (3) +NLNIP (4)

NSEP=SURFAC
NSEP1=SURFAC-NSRC
WRITE(NO,*) 'NUMBER OF SURFACES IN EXHAUST PORT = ',NSEP1
NSEPX=NSEP
```

```
C *****
C *
C * BEGIN GENERATION OF THE SURFACES OF THE *
C * INTAKE PORT *
C *
C *****
```

```
IF (IIPQUE.EQ.1) THEN
```

```
DO 780 JJ=1,IEND
```

```
SURT=SURFAC
ED1=NEDEP+1
ED2=NEDEP+28
ED3=NEDEP+15
ED4=NEDEP+29
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
DO 760 J=1,21
```

```
IF (J.EQ.1) THEN
```

```
NUMB=3
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED2,ED3,ED4
NUMB=4
```

```
ENDIF
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
IF (J.EQ.10) THEN
```

```
NUMB=3
ED2=ED2+1
ED3=ED3+1
ED4=ED4-27
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED2,ED3,ED4
ED2=ED2-28
ED3=ED3-14
ED4=ED4+14
NUMB=4
```

```
GEN47370
GEN47380
GEN47390
GEN47400
GEN47410
GEN47420
GEN47430
GEN47440
GEN47450
GEN47460
GEN47470
GEN47480
GEN47490
GEN47500
GEN47510
GEN47520
GEN47530
GEN47540
GEN47550
GEN47560
GEN47570
GEN47580
GEN47590
GEN47600
GEN47610
GEN47620
GEN47630
GEN47640
GEN47650
GEN47660
GEN47670
GEN47680
GEN47690
GEN47700
GEN47710
GEN47720
GEN47730
GEN47740
GEN47750
GEN47760
GEN47770
GEN47780
GEN47790
GEN47800
GEN47810
GEN47820
GEN47830
GEN47840
GEN47850
GEN47860
GEN47870
GEN47880
GEN47890
GEN47900
GEN47910
GEN47920
GEN47930
GEN47940
GEN47950
GEN47960
GEN47970
GEN47980
GEN47990
GEN48000
```

```

                                ENDIF
                                IF (J.EQ.11) THEN
                                    ED1=ED1+3
                                    ED2=ED2+27
                                    ED3=ED3+38
                                    ED4=ED4+13
                                ENDIF
                                IF (((JJ.EQ.3).OR.(JJ.EQ.4)).AND.(J.EQ.11)) GO TO 765
                                IF (J.EQ.16) THEN
                                    ED1=ED1+2
                                    ED2=ED2+1
                                    ED4=ED4+1
                                ENDIF
760                                CONTINUE
765    IF (JJ.EQ.IEND) GO TO 775
                                ED1=NEDEP+1
                                ED2=NEDEP+63
                                ED3=NEDEP+1+NLNIP(JJ)
                                ED4=NEDEP+64
                                IF ((JJ.EQ.3).OR.(JJ.EQ.4)) THEN
                                    ED2=ED2-22
                                    ED4=ED4-22
                                ENDIF
                                SURFAC=SURFAC+1
                                WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
                                WRITE(12,645) ED1,ED2,ED3,ED4
                                DO 770 J=1,61
                                    ED1=ED1+1
                                    ED2=ED2+1
                                    ED3=ED3+1
                                    ED4=ED4+1
                                    SURFAC=SURFAC+1
                                    WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
                                    WRITE(12,645) ED1,ED2,ED3,ED4
                                    IF (J.EQ.10) ED4=ED4-12
                                    IF (J.EQ.11) ED4=ED4+11
                                    IF (J.EQ.11) ED2=ED2-1
                                    IF (J.EQ.25) ED4=ED4-14
                                    IF (J.EQ.26) ED2=ED2-26
                                    IF (J.EQ.28) ED2=ED2-1
                                IF (((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)).AND.
#                                (J.EQ.39)) GO TO 775
                                IF (J.EQ.39) ED2=ED2+2

```

GEN48010  
GEN48020  
GEN48030  
GEN48040  
GEN48050  
GEN48060  
GEN48070  
GEN48080  
GEN48090  
GEN48100  
GEN48110  
GEN48120  
GEN48130  
GEN48140  
GEN48150  
GEN48160  
GEN48170  
GEN48180  
GEN48190  
GEN48200  
GEN48210  
GEN48220  
GEN48230  
GEN48240  
GEN48250  
GEN48260  
GEN48270  
GEN48280  
GEN48290  
GEN48300  
GEN48310  
GEN48320  
GEN48330  
GEN48340  
GEN48350  
GEN48360  
GEN48370  
GEN48380  
GEN48390  
GEN48400  
GEN48410  
GEN48420  
GEN48430  
GEN48440  
GEN48450  
GEN48460  
GEN48470  
GEN48480  
GEN48490  
GEN48500  
GEN48510  
GEN48520  
GEN48530  
GEN48540  
GEN48550  
GEN48560  
GEN48570  
GEN48580  
GEN48590  
GEN48600  
GEN48610  
GEN48620  
GEN48630  
GEN48640

```

      IF (J.EQ.45) ED2=ED2+1
      IF (J.EQ.50) ED2=ED2-14
      IF (J.EQ.51) ED2=ED2+13
      IF (J.EQ.51) ED4=ED4-11
      IF (J.EQ.56) ED2=ED2+1
      IF (J.EQ.56) ED4=ED4+1
770      CONTINUE
775      NEDEP=NEDEP+NLNIP(JJ)
      SURB=SURFAC
      NSRIP(JJ)=SURB-SURT
      IF (JJ.EQ.4) NSIP2(JJ)=SURFAC
780      CONTINUE
      NSIP1=SURFAC
      DO 815 JJ=1,2
      ED1=TEMPID+47
      ED2=NEDEP+27
      ED3=NEDEP+1
      ED4=NEDEP+23
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      ED1=ED1-25
      ED2=ED2-4
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      DO 785 J=1,2
      ED1=ED1-1
      ED2=ED2+1
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.1) ED1=ED1+26
785      CONTINUE
      ED1=NEDEP+1
      ED2=NEDEP+13
      ED3=NEDEP+5
      ED4=NEDEP+14
      SURFAC=SURFAC+1
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4
      DO 790 J=1,7
      ED1=ED1+1
      ED2=ED2+1
      ED3=ED3+1
      ED4=ED4+1
```

790

CONTINUE

IF (JJ.EQ.2) GO TO 810

```
ED1=NEDEP+1
ED2=NEDEP+28
ED3=NEDEP+1+NLNIP1(JJ)
ED4=NEDEP+29
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

DO 795 J=1,11

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

IF ((J.EQ.3).OR.(J.EQ.7)) THEN

```
ED2=ED2+1
ED4=ED4+1
```

ENDIF

795

CONTINUE

```
ED1=NEDEP+27
ED2=TEMPID+95
ED3=NEDEP+27+NLNIP1(JJ)
ED4=NEDEP+28
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

```
ED1=ED1-4
ED2=ED2-11
ED3=ED3-4
ED4=ED4+1
SURFAC=SURFAC+1
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
WRITE(12,645) ED1,ED2,ED3,ED4
```

DO 800 J=1,3

```
ED1=ED1+1
ED2=ED2-1
ED3=ED3+1
ED4=ED4+1
```

GEN49290  
GEN49300  
GEN49310  
GEN49320  
GEN49330  
GEN49340  
GEN49350  
GEN49360  
GEN49370  
GEN49380  
GEN49390  
GEN49400  
GEN49410  
GEN49420  
GEN49430  
GEN49440  
GEN49450  
GEN49460  
GEN49470  
GEN49480  
GEN49490  
GEN49500  
GEN49510  
GEN49520  
GEN49530  
GEN49540  
GEN49550  
GEN49560  
GEN49570  
GEN49580  
GEN49590  
GEN49600  
GEN49610  
GEN49620  
GEN49630  
GEN49640  
GEN49650  
GEN49660  
GEN49670  
GEN49680  
GEN49690  
GEN49700  
GEN49710  
GEN49720  
GEN49730  
GEN49740  
GEN49750  
GEN49760  
GEN49770  
GEN49780  
GEN49790  
GEN49800  
GEN49810  
GEN49820  
GEN49830  
GEN49840  
GEN49850  
GEN49860  
GEN49870  
GEN49880  
GEN49890  
GEN49900  
GEN49910  
GEN49920

C-4



```

      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.2) ED2=ED2+13
      CONTINUE
      ED1=NEDEP+13
      ED2=NEDEP+28
      ED3=NEDEP+13+NLNIP1 (JJ)
      ED4=NEDEP+33
      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      DO 805 J=1,9
      ED1=ED1+1
      ED2=ED2+1
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      CONTINUE
      NEDEP=NEDEP+NLNIP1 (JJ)
      TEMPID=TEMPID+NLNIP1 (JJ)
      CONTINUE
      DO 850 JJ=5,IEND
      SURT=SURFAC
      ED1=TEMPI1+47
      ED2=NEDEP+27
      ED3=NEDEP+1
      ED4=NEDEP+23
      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      ED1=ED1-25
      ED2=ED2-4
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      DO 820 J=1,2
      ED1=ED1-1
      ED2=ED2+1
      ED3=ED3+1
      ED4=ED4+1
      SURFAC=SURFAC+1
      WRITE (12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE (12,645) ED1,ED2,ED3,ED4
      IF (J.EQ.1) ED1=ED1+26
```

```
820      CONTINUE                                     GEN50570
                                                GEN50580
      ED1=NEDEP+1                                     GEN50590
      ED2=NEDEP+13                                    GEN50600
      ED3=NEDEP+5                                     GEN50610
      ED4=NEDEP+14                                    GEN50620
      SURFAC=SURFAC+1                                GEN50630
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB    GEN50640
      WRITE(12,645) ED1,ED2,ED3,ED4                 GEN50650
                                                GEN50660
      DO 825 J=1,7                                    GEN50670
                                                GEN50680
          ED1=ED1+1                                    GEN50690
          ED2=ED2+1                                    GEN50700
          ED3=ED3+1                                    GEN50710
          ED4=ED4+1                                    GEN50720
          SURFAC=SURFAC+1                              GEN50730
          WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN50740
          WRITE(12,645) ED1,ED2,ED3,ED4               GEN50750
                                                GEN50760
          IF (J.EQ.3) THEN                             GEN50770
                                                GEN50780
              ED2=ED2+1                                GEN50790
              ED4=ED4+1                                GEN50800
                                                GEN50810
          ENDIF                                         GEN50820
                                                GEN50830
825      CONTINUE                                     GEN50840
                                                GEN50850
      IF (JJ.EQ.IEND) GO TO 845                       GEN50860
                                                GEN50870
      ED1=NEDEP+1                                     GEN50880
      ED2=NEDEP+28                                    GEN50890
      ED3=NEDEP+1+NLNIP2(JJ)                         GEN50900
      ED4=NEDEP+29                                    GEN50910
      SURFAC=SURFAC+1                                GEN50920
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB    GEN50930
      WRITE(12,645) ED1,ED2,ED3,ED4                 GEN50940
                                                GEN50950
      DO 830 J=1,11                                   GEN50960
                                                GEN50970
          ED1=ED1+1                                    GEN50980
          ED2=ED2+1                                    GEN50990
          ED3=ED3+1                                    GEN51000
          ED4=ED4+1                                    GEN51010
          SURFAC=SURFAC+1                              GEN51020
          WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB GEN51030
          WRITE(12,645) ED1,ED2,ED3,ED4               GEN51040
                                                GEN51050
          IF ((J.EQ.3).OR.(J.EQ.7)) THEN              GEN51060
                                                GEN51070
              ED2=ED2+1                                GEN51080
              ED4=ED4+1                                GEN51090
                                                GEN51100
          ENDIF                                         GEN51110
                                                GEN51120
830      CONTINUE                                     GEN51130
                                                GEN51140
      ED1=NEDEP+27                                    GEN51150
      ED2=TEMPI1+95                                   GEN51160
      ED3=NEDEP+27+NLNIP2(JJ)                         GEN51170
      ED4=NEDEP+28                                    GEN51180
      SURFAC=SURFAC+1                                GEN51190
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB    GEN51200
```

WRITE(12,645) ED1,ED2,ED3,ED4

ED1=ED1-4

ED2=ED2-11

ED3=ED3-4

ED4=ED4+1

SURFAC=SURFAC+1

WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB

WRITE(12,645) ED1,ED2,ED3,ED4

DO 835 J=1,3

ED1=ED1+1

ED2=ED2-1

ED3=ED3+1

ED4=ED4+1

SURFAC=SURFAC+1

WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB

WRITE(12,645) ED1,ED2,ED3,ED4

- IF (J.EQ.2) ED2=ED2+13

835

CONTINUE

ED1=NEDEP+13

ED2=NEDEP+28

ED3=NEDEP+13+NLNIP2(JJ)

ED4=NEDEP+33

SURFAC=SURFAC+1

WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB

WRITE(12,645) ED1,ED2,ED3,ED4

DO 840 J=1,9

ED1=ED1+1

ED2=ED2+1

ED3=ED3+1

ED4=ED4+1

SURFAC=SURFAC+1

WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB

WRITE(12,645) ED1,ED2,ED3,ED4

840

CONTINUE

845

NEDEP=NEDEP+NLNIP2(JJ)

TEMPI1=TEMPI1+NLNIP(JJ)

SURB=SURFAC

NSIP3(JJ)=SURB-SURT

850

CONTINUE

ENDIF

NSRTL=SURFAC

NSIP5=SURFAC+1-NSRC-NSEP1

WRITE(NO,\*) 'NUMBER OF SURFACES IN INTAKE PORT = ',NSIP5

C \*\*\*\*\*  
C \*  
C \* BEGIN GENERATION OF THE SURFACES OF THE  
C \* SPARK PLUG PORT  
C \*  
C \*\*\*\*\*

GEN51210  
GEN51220  
GEN51230  
GEN51240  
GEN51250  
GEN51260  
GEN51270  
GEN51280  
GEN51290  
GEN51300  
GEN51310  
GEN51320  
GEN51330  
GEN51340  
GEN51350  
GEN51360  
GEN51370  
GEN51380  
GEN51390  
GEN51400  
GEN51410  
GEN51420  
GEN51430  
GEN51440  
GEN51450  
GEN51460  
GEN51470  
GEN51480  
GEN51490  
GEN51500  
GEN51510  
GEN51520  
GEN51530  
GEN51540  
GEN51550  
GEN51560  
GEN51570  
GEN51580  
GEN51590  
GEN51600  
GEN51610  
GEN51620  
GEN51630  
GEN51640  
GEN51650  
GEN51660  
GEN51670  
GEN51680  
GEN51690  
GEN51700  
GEN51710  
GEN51720  
GEN51730  
GEN51740  
GEN51750  
GEN51760  
GEN51770  
GEN51780  
GEN51790  
GEN51800  
GEN51810  
GEN51820  
GEN51830  
GEN51840

DO 875 JJJ=1,ISP

DO 870 JJ=1,IEND

SURT=SURFAC  
ED1=NLNTL+1  
ED2=NLNTL+36  
ED3=NLNTL+11  
ED4=NLNTL+37  
SURFAC=SURFAC+1  
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB  
WRITE(12,645) ED1,ED2,ED3,ED4

DO 855 J=1,19

ED1=ED1+1  
ED2=ED2+1  
ED3=ED3+1  
ED4=ED4+1  
SURFAC=SURFAC+1  
WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB  
WRITE(12,645) ED1,ED2,ED3,ED4

IF (J.EQ.5) ED4=ED4-34

IF (J.EQ.6) THEN

ED2=ED2-34  
ED3=ED3-8  
ED4=ED4+26

ENDIF

IF (J.EQ.7) THEN

ED1=ED1+1  
ED2=ED2+33  
ED3=ED3+7  
ED4=ED4+7

ENDIF

IF (J.EQ.14) ED4=ED4-8

IF (J.EQ.15) THEN

ED1=ED1+2  
ED2=ED2-25  
ED3=ED3+6  
ED4=ED4-17

ENDIF

IF (J.EQ.17) ED1=ED1+2  
IF (J.EQ.17) ED2=ED2+1  
IF (J.EQ.17) ED4=ED4+1

855

CONTINUE

IF (JJ.EQ.IEND) GO TO 865

ED1=NLNTL+1  
ED2=NLNTL+51

GEN51850  
GEN51860  
GEN51870  
GEN51880  
GEN51890  
GEN51900  
GEN51910  
GEN51920  
GEN51930  
GEN51940  
GEN51950  
GEN51960  
GEN51970  
GEN51980  
GEN51990  
GEN52000  
GEN52010  
GEN52020  
GEN52030  
GEN52040  
GEN52050  
GEN52060  
GEN52070  
GEN52080  
GEN52090  
GEN52100  
GEN52110  
GEN52120  
GEN52130  
GEN52140  
GEN52150  
GEN52160  
GEN52170  
GEN52180  
GEN52190  
GEN52200  
GEN52210  
GEN52220  
GEN52230  
GEN52240  
GEN52250  
GEN52260  
GEN52270  
GEN52280  
GEN52290  
GEN52300  
GEN52310  
GEN52320  
GEN52330  
GEN52340  
GEN52350  
GEN52360  
GEN52370  
GEN52380  
GEN52390  
GEN52400  
GEN52410  
GEN52420  
GEN52430  
GEN52440  
GEN52450  
GEN52460  
GEN52470  
GEN52480

```
ED3=NLNTL+1+NLNSP (JJ)
ED4=NLNTL+52
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4
```

```
DO 860 J=1, 49
```

```
ED1=ED1+1
ED2=ED2+1
ED3=ED3+1
ED4=ED4+1
SURFAC=SURFAC+1
WRITE (12, 640) SURFAC, COLOR, DASH, GRADE, NUMB
WRITE (12, 645) ED1, ED2, ED3, ED4
```

```
IF (J.EQ.6) ED4=ED4-8
```

```
IF (J.EQ.7) ED2=ED2-1
```

```
IF (J.EQ.7) ED4=ED4+7
```

```
IF (J.EQ.15) ED4=ED4-8
```

```
IF (J.EQ.16) ED4=ED4+8
```

```
IF (J.EQ.23) ED4=ED4-8
```

```
IF (J.EQ.24) ED2=ED2-6
```

```
IF (J.EQ.24) ED4=ED4+7
```

```
IF (J.EQ.27) ED2=ED2+1
```

```
IF (J.EQ.29) ED2=ED2-8
```

```
IF (J.EQ.30) ED2=ED2+7
```

```
IF (J.EQ.30) ED4=ED4-5
```

```
IF (J.EQ.32) ED2=ED2+1
```

```
IF (J.EQ.32) ED4=ED4+1
```

```
IF (J.EQ.34) ED2=ED2-29
```

```
IF (J.EQ.34) ED4=ED4-21
```

```
IF (J.EQ.41) ED2=ED2+1
```

```
860 CONTINUE
```

```
865 NLNTL=NLNTL+NLNSP (JJ)
SURB=SURFAC
NSRSP (JJ)=SURB-SURT
```

```
870 CONTINUE
```

```
875 CONTINUE
```

```
NSSP=SURFAC-NSRC-NSEP1-NSIP5+1
WRITE (NO, *) 'NUMBER OF SURFACES IN SPARK PLUG(S) = ', NSSP
```

```
ED1=ISTCN1+SPLINE
ED2=ISTCN2+SPLINE
ED3=ISTCN3+SPLINE
ED4=ISTCN4+SPLINE
SURFAC=SURFAC+1
WRITE (20, *) SURFAC, NUMBER
```

```
GEN52490
GEN52500
GEN52510
GEN52520
GEN52530
GEN52540
GEN52550
GEN52560
GEN52570
GEN52580
GEN5259%[% [ PrinterEr
GEN52600
GEN52610
GEN52620
GEN52630
GEN52640
GEN52650
GEN52660
GEN52670
GEN52680
GEN52690
GEN52700
GEN52710
GEN52720
GEN52730
GEN52740
GEN52750
GEN52760
GEN52770
GEN52780
GEN52790
GEN52800
GEN52810
GEN52820
GEN52830
GEN52840
GEN52850
GEN52860
GEN52870
GEN52880
GEN52890
GEN52900
GEN52910
GEN52920
GEN52930
GEN52940
GEN52950
GEN52960
GEN52970
GEN52980
GEN52990
GEN53000
GEN53010
GEN53020
GEN53030
GEN53040
GEN53050
GEN53060
GEN53070
GEN53080
GEN53090
GEN53100
GEN53110
GEN53120
```

```
      WRITE(12,640) SURFAC,COLOR,DASH,GRADE,NUMB
      WRITE(12,645) ED1,ED2,ED3,ED4

CLOSE (20)
WRITE(12,245)
WRITE(12,245)

WRITE(NO,*) 'TOTAL NUMBER OF SURFACES = ',SURFAC
WRITE(NO,*) ' '
WRITE(NO,*) ' '

WRITE(NO,*) 'SURFACE LABELS OF RC SEGMENTS ARE          1 TO',NSRC
WRITE(NO,*) 'SURFACE LBLs OF EXHST PORT ARE',NSRC+1,'TO',NSEP1+NSRC
WRITE(NO,*) 'SURFACE LBLs OF INTKE PORT ARE',NSRC+NSEP1+1,'TO',NSIP
#5+NSRC+NSEP1
WRITE(NO,*) 'SURFACE LBLs OF SPRK PLUGS ARE',NSRC+NSEP1+NSIP5+1,'TO
#',NSSP+NSRC+NSEP1+NSIP5
WRITE(NO,*) ' '
WRITE(NO,*) ' '

*****
*
*   END GENERATION OF THE SURFACES IN UNIVERSAL FORMAT
*
*****

*****
*
*   BEGIN GENERATION OF THE VOLUMES OF THE INNER AND
*   OUTER SHELLS IN UNIVERSAL FORMAT
*
*****

      WRITE(12,880)
880  FORMAT(4X,'39')

      SURFAC=TEMPSU
      NUMB=6
      VOLUME=1
      SUR1=1
      SUR2=6
      SUR3=SURFAC-3
      IF (CNLTYP (NUMBER) .EQ.2) SUR3=SUR3-1
      SUR4=9
      SUR5=12
      SUR6=2

      WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
885  FORMAT(5I10)
      WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
890  FORMAT(6I10)

      DO 900 I=1,NUMBER

          IF (RIBTYP (I) .EQ.1) THEN

              SUR1=SUR1+1
              SUR2=SUR2+1
              SUR3=SUR3+1
              SUR4=SUR4+1
              SUR5=SUR5+1
              SUR6=SUR6+1
              VOLUME=VOLUME+1
```

```
IF (I.NE.IVOLC(I)) THEN
    WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
    WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
ENDIF

SUR1=SUR1+2
SUR2=SUR2+1
SUR3=SUR3+2
SUR4=SUR4+1
SUR5=SUR5+2
SUR6=SUR6+2
VOLUME=VOLUME+1

IF (I.NE.IVOLC(I)) THEN
    WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
    WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
ENDIF

SUR1=SUR1+12
SUR2=SUR2+13
IF(I.EQ.1) THEN
    SUR3=SUR3+12-SURFAC
ELSE IF(CNLTYP(I-1).EQ.1) THEN
    SUR3=SUR3+12
ELSE IF(CNLTYP(I-1).EQ.2) THEN
    SUR3=SUR3+13
ENDIF

SUR4=SUR4+14
SUR5=SUR5+14
SUR6=SUR6+12
VOLUME=VOLUME+1

IF (I.NE.IVOLR(I)) THEN
    WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
    WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
ENDIF

DO 895 J=1,3

    SUR1=SUR1+1
    SUR2=SUR2+1
    SUR3=SUR3+1
    SUR4=SUR4+1
    SUR5=SUR5+1
    SUR6=SUR6+1
    VOLUME=VOLUME+1

    IF (I.NE.IVOLR(I)) THEN
        WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
        WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
    ENDIF
```

GEN53770  
GEN53780  
GEN53790  
GEN53800  
GEN53810  
GEN53820  
GEN53830  
GEN53840  
GEN53850  
GEN53860  
GEN53870  
GEN53880  
GEN53890  
GEN53900  
GEN53910  
GEN53920  
GEN53930  
GEN53940  
GEN53950  
GEN53960  
GEN53970  
GEN53980  
GEN53990  
GEN54000  
GEN54010  
GEN54020  
GEN54030  
GEN54040  
GEN54050  
GEN54060  
GEN54070  
GEN54080  
GEN54090  
GEN54100  
GEN54110  
GEN54120  
GEN54130  
GEN54140  
GEN54150  
GEN54160  
GEN54170  
GEN54180  
GEN54190  
GEN54200  
GEN54210  
GEN54220  
GEN54230  
GEN54240  
GEN54250  
GEN54260  
GEN54270  
GEN54280  
GEN54290  
GEN54300  
GEN54310  
GEN54320  
GEN54330  
GEN54340  
GEN54350  
GEN54360  
GEN54370  
GEN54380  
GEN54390  
GEN54400

ENDIF

CONTINUE

895

ELSE IF (RIBTYP(I).EQ.2) THEN

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+1

SUR6=SUR6+1

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+2

SUR2=SUR2+1

SUR3=SUR3+1

IF ((CNLTYP(I-1).EQ.2).OR.(RIBTYP(I-1).EQ.1)) SUR3=SUR3+1

IF (RIBTYP(I-1).EQ.2) SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+2

SUR6=SUR6+2

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+12

SUR2=SUR2+14

SUR3=SUR3+12

IF (CNLTYP(I-1).EQ.2) SUR3=SUR3+1

SUR4=SUR4+15

SUR5=SUR5+15

SUR6=SUR6+12

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+1

SUR6=SUR6+1

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+2

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+1

SUR6=SUR6+2

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

GEN54410

GEN54420

GEN54430

GEN54440

GEN54450

GEN54460

GEN54470

GEN54480

GEN54490

GEN54500

GEN54510

GEN54520

GEN54530

GEN54540

GEN54550

GEN54560

GEN54570

GEN54580

GEN54590

GEN54600

GEN54610

GEN54620

GEN54630

GEN54640

GEN54650

GEN54660

GEN54670

GEN54680

GEN54690

GEN54700

GEN54710

GEN54720

GEN54730

GEN54740

GEN54750

GEN54760

GEN54770

GEN54780

GEN54790

GEN54800

GEN54810

GEN54820

GEN54830

GEN54840

GEN54850

GEN54860

GEN54870

GEN54880

GEN54890

GEN54900

GEN54910

GEN54920

GEN54930

GEN54940

GEN54950

GEN54960

GEN54970

GEN54980

GEN54990

GEN55000

GEN55010

GEN55020

GEN55030

GEN55040



```

      SUR5=SUR5+1
      SUR6=SUR6+1
      VOLUME=VOLUME+1
      WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
      WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
      ENDIF
      IF(RIBTYP(I).EQ.1) THEN
        IF(CNLTYP(I).EQ.1) THEN
          IF(I.EQ.NUMBER) GO TO 900
          SUR1=SUR1+14
          SUR2=SUR2+14
          SUR3=SUR3+14
          SUR4=SUR4+13
          SUR5=SUR5+12
          SUR6=SUR6+14
          VOLUME=VOLUME+1
          IF (I.NE.IVOLC(I+1)-1) THEN
            WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
            WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
          ENDIF
        ELSE IF(CNLTYP(I).EQ.2) THEN
          IF(I.EQ.NUMBER) GO TO 900
          SUR1=SUR1+15
          SUR2=SUR2+15
          SUR3=SUR3+14
          SUR4=SUR4+14
          SUR5=SUR5+13
          SUR6=SUR6+15
          VOLUME=VOLUME+1
          IF (I.NE.IVOLC(I+1)-1) THEN
            WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
            WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
          ENDIF
        ENDIF
      ELSE IF(RIBTYP(I).EQ.2) THEN
        IF(I.EQ.NUMBER) GO TO 900
        SUR1=SUR1+14
        SUR2=SUR2+14
        SUR3=SUR3+15
        SUR4=SUR4+13
        SUR5=SUR5+12
        SUR6=SUR6+14
        VOLUME=VOLUME+1
        WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
        WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

```

```

                                ENDIF
900  CONTINUE
                                NVRC=VOLUME
                                WRITE(NO,*) 'NUMBER OF VOLUMES IN RC SEGMENTS  = ',NVRC-14
C   *****
C   *
C   *          BEGIN GENERATION OF THE VOLUMES OF THE          *
C   *          EXHAUST PORT IN UNIVERSAL FORMAT                *
C   *
C   *****
                                IF (IEPQUE.EQ.1) THEN
                                    DO 915 JJ=1,IEND-1
                                        SUR1=TEMPSU+1
                                        SUR2=TEMPSU+21
                                        SUR3=TEMPSU+46
                                        SUR4=TEMPSU+35
                                        SUR5=TEMPSU+47
                                        SUR6=TEMPSU+1+NSREP (JJ)
                                            IF ((JJ.EQ.3).OR.(JJ.EQ.4)) THEN
                                                SUR2=SUR2-8
                                                SUR3=SUR3-8
                                                SUR4=SUR4-8
                                                SUR5=SUR5-8
                                            ENDIF
                                        VOLUME=VOLUME+1
                                        WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
                                        WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
                                            DO 905 J=1,19
                                                SUR1=SUR1+1
                                                SUR2=SUR2+1
                                                SUR3=SUR3+1
                                                SUR4=SUR4+1
                                                SUR5=SUR5+1
                                                SUR6=SUR6+1
                                                VOLUME=VOLUME+1
                                                WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
                                                WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
                                                    IF (J.EQ.9) SUR5=SUR5-24
                                                    IF (J.EQ.10) THEN
                                                        SUR3=SUR3-24
                                                        SUR4=SUR4-12
                                                        SUR5=SUR5+12
                                                    ENDIF
                                                IF(((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)).AND.(J.EQ.11)) GO TO 910
                                                    IF (J.EQ.11) THEN

```

```

GEN55690
GEN55700
GEN55710
GEN55720
GEN55730
GEN55740
GEN55750
GEN55760
GEN55770
GEN55780
GEN55790
GEN55800
GEN55810
GEN55820
GEN55830
GEN55840
GEN55850
GEN55860
GEN55870
GEN55880
GEN55890
GEN55900
GEN55910
GEN55920
GEN55930
GEN55940
GEN55950
GEN55960
GEN55970
GEN55980
GEN55990
GEN56000
GEN56010
GEN56020
GEN56030
GEN56040
GEN56050
GEN56060
GEN56070
GEN56080
GEN56090
GEN56100
GEN56110
GEN56120
GEN56130
GEN56140
GEN56150
GEN56160
GEN56170
GEN56180
GEN56190
GEN56200
GEN56210
GEN56220
GEN56230
GEN56240
GEN56250
GEN56260
GEN56270
GEN56280
GEN56290
GEN56300
GEN56310
GEN56320

```

```

SUR2=SUR2+3
SUR3=SUR3+23
SUR4=SUR4+32
SUR5=SUR5+11
ENDIF
IF (J.EQ.15) THEN
SUR2=SUR2+2
SUR3=SUR3+1
SUR5=SUR5+1
ENDIF
905 CONTINUE
910 TEMPSU=TEMPSU+NSREP (JJ)
915 CONTINUE
NIMVEP=VOLUME+1
SUR1=TEMSU1+1
SUR2=TEMSU1+13
SUR3=TEMSU1+25
SUR4=TEMSU1+17
SUR5=TEMSU1+26
SUR6=TEMSU1+40
VOLUME=VOLUME+1
WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB
WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
DO 920 J=1,11
SUR1=SUR1+1
SUR2=SUR2+1
SUR3=SUR3+1
SUR4=SUR4+1
SUR5=SUR5+1
SUR6=SUR6+1
VOLUME=VOLUME+1
WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB
WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
IF (J.EQ.3) SUR3=SUR3+1
IF (J.EQ.3) SUR5=SUR5+1
IF (J.EQ.7) THEN
SUR2=TEMSU+65
SUR3=SUR3+1
SUR4=SUR4-12
SUR5=SUR5+1
ENDIF
IF (J.EQ.8) SUR2=SUR2-33
IF (J.EQ.10) SUR2=SUR2+21
920 CONTINUE
ENDIF
```

GEN56330  
GEN56340  
GEN56350  
GEN56360  
GEN56370  
GEN56380  
GEN56390  
GEN56400  
GEN56410  
GEN56420  
GEN56430  
GEN56440  
GEN56450  
GEN56460  
GEN56470  
GEN56480  
GEN56490  
GEN56500  
GEN56510  
GEN56520  
GEN56530  
GEN56540  
GEN56550  
GEN56560  
GEN56570  
GEN56580  
GEN56590  
GEN56600  
GEN56610  
GEN56620  
GEN56630  
GEN56640  
GEN56650  
GEN56660  
GEN56670  
GEN56680  
GEN56690  
GEN56700  
GEN56710  
GEN56720  
GEN56730  
GEN56740  
GEN56750  
GEN56760  
GEN56770  
GEN56780  
GEN56790  
GEN56800  
GEN56810  
GEN56820  
GEN56830  
GEN56840  
GEN56850  
GEN56860  
GEN56870  
GEN56880  
GEN56890  
GEN56900  
GEN56910  
GEN56920  
GEN56930  
GEN56940  
GEN56950  
GEN56960

NVEP=VOLUME-NVRC

WRITE(NO,\*) 'NUMBER OF VOLUMES IN EXHAUST PORT = ',NVEP

```
C *****
C *
C *          BEGIN GENERATION OF THE VOLUMES OF THE
C *          INTAKE PORT IN UNIVERSAL FORMAT
C *
C *****
```

IF (IIPQUE.EQ.1) THEN

DO 935 JJ=1,IEND-1

```
SUR1=NSEP+1
SUR2=NSEP+25
SUR3=NSEP+52
SUR4=NSEP+39
SUR5=NSEP+53
SUR6=NSEP+1+NSRIP (JJ)
```

IF ((JJ.EQ.3).OR.(JJ.EQ.4)) THEN

```
SUR2=SUR2-10
SUR3=SUR3-10
SUR4=SUR4-10
SUR5=SUR5-10
```

ENDIF

VOLUME=VOLUME+1

WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

DO 925 J=1,21

IF (J.EQ.1) THEN

```
NUMB=5
SUR1=SUR1+1
SUR3=SUR3+1
SUR4=SUR4+1
SUR5=SUR5+1
SUR6=SUR6+1
VOLUME=VOLUME+1
WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
WRITE(12,890) SUR1,SUR3,SUR4,SUR5,SUR6
NUMB=6
```

ENDIF

```
SUR1=SUR1+1
SUR2=SUR2+1
SUR3=SUR3+1
SUR4=SUR4+1
SUR5=SUR5+1
SUR6=SUR6+1
VOLUME=VOLUME+1
WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
```

IF (J.EQ.10) THEN

GEN56970  
GEN56980  
GEN56990  
GEN57000  
GEN57010  
GEN57020  
GEN57030  
GEN57040  
GEN57050  
GEN57060  
GEN57070  
GEN57080  
GEN57090  
GEN57100  
GEN57110  
GEN57120  
GEN57130  
GEN57140  
GEN57150  
GEN57160  
GEN57170  
GEN57180  
GEN57190  
GEN57200  
GEN57210  
GEN57220  
GEN57230  
GEN57240  
GEN57250  
GEN57260  
GEN57270  
GEN57280  
GEN57290  
GEN57300  
GEN57310  
GEN57320  
GEN57330  
GEN57340  
GEN57350  
GEN57360  
GEN57370  
GEN57380  
GEN57390  
GEN57400  
GEN57410  
GEN57420  
GEN57430  
GEN57440  
GEN57450  
GEN57460  
GEN57470  
GEN57480  
GEN57490  
GEN57500  
GEN57510  
GEN57520  
GEN57530  
GEN57540  
GEN57550  
GEN57560  
GEN57570  
GEN57580  
GEN57590  
GEN57600

```
NUMB=5
SUR1=SUR1+1
SUR3=SUR3+1
SUR4=SUR4+1
SUR5=SUR5-27
SUR6=SUR6+1
VOLUME=VOLUME+1
WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
WRITE(12,890) SUR1,SUR3,SUR4,SUR5,SUR6
NUMB=6

      SUR3=SUR3-28
      SUR4=SUR4-14
      SUR5=SUR5+14

ENDIF

IF(( (JJ.EQ.2) .OR. (JJ.EQ.3) .OR. (JJ.EQ.4) ) .AND. (J.EQ.11) ) GO TO 930

      IF (J.EQ.11) THEN

            SUR2=SUR2+3
            SUR3=SUR3+27
            SUR4=SUR4+38
            SUR5=SUR5+13
      ENDIF

      IF (J.EQ.16) THEN

            SUR2=SUR2+2
            SUR3=SUR3+1
            SUR5=SUR5+1
      ENDIF

925      CONTINUE

930      NSEP=NSIP+NSRIP(JJ)

935      CONTINUE

      NSEP=NSIPX
      NIMVIP=VOLUME+1

      SUR1=NSIP1+1
      SUR2=NSIP1+13
      SUR3=NSIP1+25
      SUR4=NSIP1+71
      SUR5=NSIP1+26
      SUR6=NSIP1+40
      VOLUME=VOLUME+1
      WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
      WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

      SUR1=SUR1+1
      SUR2=SUR2+1
      SUR3=SUR3+1
      SUR4=SUR4-25
      SUR5=SUR5+1
      SUR6=SUR6+1
      VOLUME=VOLUME+1
      WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
      WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
```

SUR4=SUR4-2

DO 940 J=1,10

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+1

SUR6=SUR6+1

VOLUME=VOLUME+1

WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

IF (J.EQ.1) SUR4=SUR4+24

IF (J.EQ.2) SUR3=SUR3+1

IF (J.EQ.2) SUR4=NSIP1+12

IF (J.EQ.2) SUR5=SUR5+1

IF (J.EQ.6) SUR3=SUR3+1

IF (J.EQ.6) SUR5=SUR5+1

940

CONTINUE

NSIP1=NSIP1+51

DO 950 JJ=5,IEND-1

SUR1=NSIP1+1

SUR2=NSIP1+13

SUR3=NSIP1+25

SUR4=NSIP2 (JJ-1)+71

SUR5=NSIP1+26

SUR6=NSIP1+40

VOLUME=VOLUME+1

WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4-25

SUR5=SUR5+1

SUR6=SUR6+1

VOLUME=VOLUME+1

WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

SUR4=SUR4-2

DO 945 J=1,10

SUR1=SUR1+1

SUR2=SUR2+1

SUR3=SUR3+1

SUR4=SUR4+1

SUR5=SUR5+1

SUR6=SUR6+1

VOLUME=VOLUME+1

WRITE (12,885) VOLUME,COLOR,DASH,GRADE,NUMB

WRITE (12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6

IF (J.EQ.1) SUR4=SUR4+24

GEN58250  
GEN58260  
GEN58270  
GEN58280  
GEN58290  
GEN58300  
GEN58310  
GEN58320  
GEN58330  
GEN58340  
GEN58350  
GEN58360  
GEN58370  
GEN58380  
GEN58390  
GEN58400  
GEN58410  
GEN58420  
GEN58430  
GEN58440  
GEN58450  
GEN58460  
GEN58470  
GEN58480  
GEN58490  
GEN58500  
GEN58510  
GEN58520  
GEN58530  
GEN58540  
GEN58550  
GEN58560  
GEN58570  
GEN58580  
GEN58590  
GEN58600  
GEN58610  
GEN58620  
GEN58630  
GEN58640  
GEN58650  
GEN58660  
GEN58670  
GEN58680  
GEN58690  
GEN58700  
GEN58710  
GEN58720  
GEN58730  
GEN58740  
GEN58750  
GEN58760  
GEN58770  
GEN58780  
GEN58790  
GEN58800  
GEN58810  
GEN58820  
GEN58830  
GEN58840  
GEN58850  
GEN58860  
GEN58870  
GEN58880

```

          IF (J.EQ.2) SUR3=SUR3+1
          IF (J.EQ.2) SUR4=NSIP1+12
          IF (J.EQ.2) SUR5=SUR5+1
          IF (J.EQ.6) SUR3=SUR3+1
          IF (J.EQ.6) SUR5=SUR5+1
945      CONTINUE
          NSIP1=NSIP1+NSIP3(JJ)
          NSIP2(JJ)=NSIP2(JJ-1)+NSRIP(JJ)
950      CONTINUE
      ENDIF
      NVIP=VOLUME-NVRC-NVEP
      WRITE(NO,*) 'NUMBER OF VOLUMES IN INTAKE PORT   = ',NVIP
      *****
      *
      *          BEGIN GENERATION OF THE VOLUMES OF THE
      *          SPARK PLUG IN UNIVERSAL FORMAT
      *
      *****
      DO 965 JJJ=1,ISP
          NIVSP2=VOLUME+1
          DO 960 JJ=1,IEND-1
              SUR1=NSRTL+1
              SUR2=NSRTL+21
              SUR3=NSRTL+56
              SUR4=NSRTL+31
              SUR5=NSRTL+57
              SUR6=NSRTL+1+NSRSP(JJ)
              VOLUME=VOLUME+1
              WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
              WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
          DO 955 J=1,19
              SUR1=SUR1+1
              SUR2=SUR2+1
              SUR3=SUR3+1
              SUR4=SUR4+1
              SUR5=SUR5+1
              SUR6=SUR6+1
              VOLUME=VOLUME+1
              WRITE(12,885) VOLUME,COLOR,DASH,GRADE,NUMB
              WRITE(12,890) SUR1,SUR2,SUR3,SUR4,SUR5,SUR6
              IF (J.EQ.5) SUR5=SUR5-34
              IF (J.EQ.6) THEN
                  SUR3=SUR3-34
                  SUR4=SUR4-8
                  SUR5=SUR5+26

```

```

GEN58890
GEN58900
GEN58910
GEN58920
GEN58930
GEN58940
GEN58950
GEN58960
GEN58970
GEN58980
GEN58990
GEN59000
GEN59010
GEN59020
GEN59030
GEN59040
GEN59050
GEN59060
GEN59070
GEN59080
GEN59090
GEN59100
GEN59110
GEN59120
GEN59130
GEN59140
GEN59150
GEN59160
GEN59170
GEN59180
GEN59190
GEN59200
GEN59210
GEN59220
GEN59230
GEN59240
GEN59250
GEN59260
GEN59270
GEN59280
GEN59290
GEN59300
GEN59310
GEN59320
GEN59330
GEN59340
GEN59350
GEN59360
GEN59370
GEN59380
GEN59390
GEN59400
GEN59410
GEN59420
GEN59430
GEN59440
GEN59450
GEN59460
GEN59470
GEN59480
GEN59490
GEN59500
GEN59510
GEN59520

```

```
ENDIF
IF (J.EQ.7) THEN
    SUR2=SUR2+1
    SUR3=SUR3+33
    SUR4=SUR4+7
    SUR5=SUR5+7
ENDIF
IF (J.EQ.14) SUR5=SUR5-8
IF (J.EQ.15) THEN
    SUR2=SUR2+2
    SUR3=SUR3-25
    SUR4=SUR4+6
    SUR5=SUR5-17
ENDIF
IF (J.EQ.17) THEN
    SUR2=SUR2+2
    SUR3=SUR3+1
    SUR5=SUR5+1
ENDIF
CONTINUE
955      NSRTL=NSRTL+NSRSP (JJ)
960      CONTINUE
IF (JJ.EQ.IEND) NSRTL=NSRTL+NSRSP (IEND)
965      CONTINUE
NVSP=VOLUME-NVRC-NVEP-NVIP
WRITE(NO,*) 'NUMBER OF VOLUMES IN SPARK PLUG(S) = ',NVSP
WRITE(12,245)
WRITE(NO,*) 'TOTAL NUMBER OF VOLUMES = ',VOLUME-14
WRITE(NO,*) ' '
WRITE(NO,*) ' '
WRITE(NO,*) 'VOLUME LABELS OF RC SEGMENTS ARE 1 TO',NVRC
WRITE(NO,*) 'VOLUME LBLs OF EXHST PORT ARE ',NVRC+1,'TO',NVEP+NVRC
WRITE(NO,*) 'VOLUME LBLs OF INTKE PORT ARE ',NVRC+NVEP+1,'TO',NVIP
# +NVRC+NVEP
WRITE(NO,*) 'VOLUME LBLs OF SPRK PLUGS ARE ',NVRC+NVEP+NVIP+1,'TO'
#,NVSP+NVRC+NVEP+NVIP
WRITE(NO,*) ' '
WRITE(NO,*) ' '
WRITE(13,970) NVRC+1,NIMVEP,NVRC+NVEP+1,NIMVIP
970      FORMAT(4I4)
WRITE(13,970) NVRC+NVEP+NVIP+1,NIVSP2,NVSP+NVRC+NVEP+NVIP,IEND
WRITE(13,*) TCHNL,CCC,ISRCT,ISP
CLOSE (13)
```

GEN59530  
GEN59540  
GEN59550  
GEN59560  
GEN59570  
GEN59580  
GEN59590  
GEN59600  
GEN59610  
GEN59620  
GEN59630  
GEN59640  
GEN59650  
GEN59660  
GEN59670  
GEN59680  
GEN59690  
GEN59700  
GEN59710  
GEN59720  
GEN59730  
GEN59740  
GEN59750  
GEN59760  
GEN59770  
GEN59780  
GEN59790  
GEN59800  
GEN59810  
GEN59820  
GEN59830  
GEN59840  
GEN59850  
GEN59860  
GEN59870  
GEN59880  
GEN59890  
GEN59900  
GEN59910  
GEN59920  
GEN59930  
GEN59940  
GEN59950  
GEN59960  
GEN59970  
GEN59980  
GEN59990  
GEN60000  
GEN60010  
GEN60020  
GEN60030  
GEN60040  
GEN60050  
GEN60060  
GEN60070  
GEN60080  
GEN60090  
GEN60100  
GEN60110  
GEN60120  
GEN60130  
GEN60140  
GEN60150  
GEN60160



```

WRITE(NO,975)
975 FORMAT('0','ALL GEOMETRIC ENTITIES (POINTS THROUGH VOLUMES)')
WRITE(NO,980)
980 FORMAT('0','HAVE BEEN GENERATED. THE PROGRAM CAN BE EXITED AT')
WRITE(NO,985)
985 FORMAT('0','THIS POINT IF MANUAL ELEMENT GENERATION IS DESIRED.')
WRITE(NO,990)
990 FORMAT('0','ENTER A ...')
WRITE(NO,995)
995 FORMAT('0','      1 TO CONTINUE MODEL GENERATION OR ...')
WRITE(NO,999)
999 FORMAT('0','      2 TO STOP MODEL GENERATION.')

READ(5,*) ICONTI
IF (STATUS.NE.0) WRITE(8,*) ICONTI
WRITE(NO,*) ICONTI

IF(ICONTI.EQ.2) THEN

  PRINT*, ' '
  PRINT*, ' THE UNIVERSAL HAS BEEN CREATED.'
  PRINT*, ' '
  PRINT*, ' BEFORE THE UNIVERSAL FILE CAN BE READ,'
  PRINT*, ' THE USER MUST ENTER THE SDRC SOFTWARE.'

  PRINT*, ' '
  PRINT*, ' NOTE THAT A FILE CALLED "MODEL DATA" CAN NOT'
  PRINT*, ' EXIST ON YOUR DISK. IF IT DOES, AN ERROR WILL'
  PRINT*, ' RESULT WHEN YOU ENTER THE SDRC SOFTWARE. TO'
  PRINT*, ' CORRECT THE ERROR, CHANGE THE NAME OF THE'
  PRINT*, ' "MODEL DATA" FILE THAT EXISTS ON YOUR DISK'
  PRINT*, ' TO A DIFFERENT NAME.'
  PRINT*, ' '
  PRINT*, ' AFTER ENTERING THE SDRC SOFTWARE IN THE PROGRAM MODE,'
  PRINT*, ' RESPOND "R" (FOR RUN) TO THE FIRST QUESTION,'
  PRINT*, ' RESPOND "GO" TO THE SECOND QUESTION AND'
  PRINT*, ' IN RESPONSE TO THE THIRD QUESTION,'
  PRINT*, ' ENTER THE TERMINAL TYPE THAT YOU ARE USING.'

  CALL GO (ICONTI)

  STOP
ENDIF

RETURN
END

SUBROUTINE MNE (NO)

  DIMENSION PHIONE(50),PHITWO(0:50),JJJ(50),JJJC(50),JJJR(50),
#      IRIB(50),JJJSC(50),NUMES(20),JSUR(20)

  CHARACTER *10 NOAL,SLASH,GENERL,YES

  NOAL='4 -1NOAL'
  SLASH='1 -1/'
  GENERL='10 -1'
  YES='3 -1YES'

  *****
C *
C * READ VALUES THAT WERE CALCULATED IN THE MAIN PROGRAM *
C *
C *****

```

GEN60170  
 GEN60180  
 GEN60190  
 GEN60200  
 GEN60210  
 GEN60220  
 GEN60230  
 GEN60240  
 GEN60250  
 GEN60260  
 GEN60270  
 GEN60280  
 GEN60290  
 GEN60300  
 GEN60310  
 GEN60320  
 GEN60330  
 GEN60340  
 GEN60350  
 GEN60360  
 GEN60370  
 GEN60380  
 GEN60390  
 GEN60400  
 GEN60410  
 GEN60420  
 GEN60430  
 GEN60440  
 GEN60450  
 GEN60460  
 GEN60470  
 GEN60480  
 GEN60490  
 GEN60500  
 GEN60510  
 GEN60520  
 GEN60530  
 GEN60540  
 GEN60550  
 GEN60560  
 GEN60570  
 GEN60580  
 GEN60590  
 GEN60600  
 GEN60610  
 GEN60620  
 GEN60630  
 GEN60640  
 GEN60650  
 GEN60660  
 GEN60670  
 GEN60680  
 GEN60690  
 GEN60700  
 GEN60710  
 GEN60720  
 GEN60730  
 GEN60740  
 GEN60750  
 GEN60760  
 GEN60770  
 GEN60780  
 GEN60790  
 GEN60800

```
OPEN (13)
OPEN (20)
```

```
READ(20,*) NUMBER
READ(13,*) PI,R,TRANS
```

```
DO 15 J=1,NUMBER
```

```
      READ(13,*) JJJ(J)
      READ(13,5) PHIONE(J),PHITWO(J)
5      FORMAT(F13.5,5X,F13.5)
      READ(13,*) IRIB(J)
10     READ(13,10) JJJC(J),JJJR(J)
      FORMAT(F13.5,5X,F13.5)
```

```
15     CONTINUE
```

```
      READ(13,*) NIVEP,NIMVEP,NIVIP,NIMVIP
      READ(13,*) NIVSP1,NIVSP2,NFVSP,IEND
      READ(13,*) TCHNL,CCC,NSTCN,ISP
      ISUR=1
      MN=0
      NTW=2
```

```
      DO 20 III=1,NSTCN+1
```

```
        READ(20,*) JJJSC(III),JSUR(III)
```

```
20     CONTINUE
```

```
C *****
C *
C *                               END READ
C *
C *****
IF (CCC.GT.0.0) THEN
```

```
      I=1
      ISTOP=6
      ISTOP1=36
      ISTOP2=12
      ISTOP3=14
```

```
ELSE
```

```
      I=2
      ISTOP=4
      ISTOP1=24
      ISTOP2=10
      ISTOP3=11
```

```
ENDIF
```

```
C *****
C *
C *      SET ELEMENT TYPE FOR GENERATION OF THICK SHELLS
C *
C *****
```

```
      PHITWO(0)=ATAN((TRANS-TCHNL)/R)
```

```
      WRITE(15,25) GENERL
```

GEN60810  
GEN60820  
GEN60830  
GEN60840  
GEN60850  
GEN60860  
GEN60870  
GEN60880  
GEN60890  
GEN60900  
GEN60910  
GEN60920  
GEN60930  
GEN60940  
GEN60950  
GEN60960  
GEN60970  
GEN60980  
GEN60990  
GEN61000  
GEN61010  
GEN61020  
GEN61030  
GEN61040  
GEN61050  
GEN61060  
GEN61070  
GEN61080  
GEN61090  
GEN61100  
GEN61110  
GEN61120  
GEN61130  
GEN61140  
GEN61150  
GEN61160  
GEN61170  
GEN61180  
GEN61190  
GEN61200  
GEN61210  
GEN61220  
GEN61230  
GEN61240  
GEN61250  
GEN61260  
GEN61270  
GEN61280  
GEN61290  
GEN61300  
GEN61310  
GEN61320  
GEN61330  
GEN61340  
GEN61350  
GEN61360  
GEN61370  
GEN61380  
GEN61390  
GEN61400  
GEN61410  
GEN61420  
GEN61430  
GEN61440

```
25      FORMAT(1X,A5,'AU')
30      WRITE(15,30) SLASH
      FORMAT(2X,A5)
35      WRITE(15,35) GENERL
      FORMAT(1X,A5,'T')
40      WRITE(15,40) GENERL
      FORMAT(1X,A5,'MA')
42      WRITE(15,42) GENERL
      FORMAT(1X,A5,'EL')
44      WRITE(15,44) GENERL
      FORMAT(1X,A5,'DEF')
46      WRITE(15,46) GENERL
      FORMAT(1X,A5,'TY')
48      WRITE(15,48) GENERL
      FORMAT(1X,A5,'TK')
50      WRITE(15,50) GENERL
      FORMAT(1X,A5,'PB')
52      WRITE(15,52) SLASH
      FORMAT(2X,A5)
54      WRITE(15,54) GENERL
      FORMAT(1X,A5,'MD')
56      WRITE(15,56) GENERL
      FORMAT(1X,A5,'CR')
58      WRITE(15,58)
      FORMAT(2X,'4 27 K')

      DO 195 J=1,NUMBER
          DO 190 JJ=1,ISTOP
C *****
C *
C *      CALCUALTE THE LENGTH OF THE RIBS AND CHANNELS
C *
C *****
      IF ((JJJ(J-1).EQ.3).AND.(JJJ(J).EQ.2)) THEN
          IF ((JJ.LT.4).AND.(CCC.GT.0.0)) THEN
              PHIT=PHITWO(J)*PI/180.0
              Y=R*SIN(PHIT)+TRANS
              PHI=ABS(ABS(Y)-ABS(PHIONE(J)))
          ELSE IF ((JJ.GT.3).AND.(CCC.GT.0.0)) THEN
              PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J)))
          ENDIF
          IF ((JJ.LT.3).AND.(CCC.EQ.0.0)) THEN
```

```
GEN61450
GEN61460
GEN61470
GEN61480
GEN61490
GEN61500
GEN61510
GEN61520
GEN61530
GEN61540
GEN61550
GEN61560
GEN61570
GEN61580
GEN61590
GEN61600
GEN61610
GEN61620
GEN61630
GEN61640
GEN61650
GEN61660
GEN61670
GEN61680
GEN61690
GEN61700
GEN61710
GEN61720
GEN61730
GEN61740
GEN61750
GEN61760
GEN61770
GEN61780
GEN61790
GEN61800
GEN61810
GEN61820
GEN61830
GEN61840
GEN61850
GEN61860
GEN61870
GEN61880
GEN61890
GEN61900
GEN61910
GEN61920
GEN61930
GEN61940
GEN61950
GEN61960
GEN61970
GEN61980
GEN61990
GEN62000
GEN62010
GEN62020
GEN62030
GEN62040
GEN62050
GEN62060
GEN62070
GEN62080
```

```
      PHIT=PHITWO(J)*PI/180.0
      Y=R*SIN(PHIT)+TRANS
      PHI=ABS(ABS(Y)-ABS(PHIONE(J)))
      ELSE IF ((JJ.GT.2).AND.(CCC.EQ.0.0)) THEN
        PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J)))
      ENDIF
    ELSE IF ((JJJ(J).EQ.3).AND.(JJJ(J-1).EQ.2)) THEN
      IF ((JJ.LT.4).AND.(CCC.GT.0.0)) THEN
        PHIT=PHIONE(J)*PI/180.0
        Y=R*SIN(PHIT)+TRANS
        PHI=ABS(ABS(Y)-ABS(PHITWO(J-1)))
      ELSE IF ((JJ.GT.3).AND.(CCC.GT.0.0)) THEN
        PHI=ABS(PHITWO(J)-PHIONE(J))
      ENDIF
      IF ((JJ.LT.3).AND.(CCC.EQ.0.0)) THEN
        PHIT=PHIONE(J)*PI/180.0
        Y=R*SIN(PHIT)+TRANS
        PHI=ABS(ABS(Y)-ABS(PHITWO(J-1)))
      ELSE IF ((JJ.GT.2).AND.(CCC.EQ.0.0)) THEN
        PHI=ABS(PHITWO(J)-PHIONE(J))
      ENDIF
    ELSE IF (JJJ(J).EQ.2) THEN
      IF ((JJ.LT.4).AND.(CCC.GT.0.0)) THEN
        PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J-1)))
      ELSE IF ((JJ.GT.3).AND.(CCC.GT.0.0)) THEN
        PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J)))
      ENDIF
      IF ((JJ.LT.3).AND.(CCC.EQ.0.0)) THEN
        PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J-1)))
      ELSE IF ((JJ.GT.2).AND.(CCC.EQ.0.0)) THEN
        PHI=ABS(ABS(PHIONE(J))-ABS(PHITWO(J)))
      ENDIF
    ELSE IF ((JJJ(J).EQ.1).OR.(JJJ(J).EQ.3)) THEN
      IF ((JJ.LT.4).AND.(CCC.GT.0.0)) THEN
        PHI=ABS(PHIONE(J)-PHITWO(J-1))
```

```
GEN62090
GEN62100
GEN62110
GEN62120
GEN62130
GEN62140
GEN62150
GEN62160
GEN62170
GEN62180
GEN62190
GEN62200
GEN62210
GEN62220
GEN62230
GEN62240
GEN62250
GEN62260
GEN62270
GEN62280
GEN62290
GEN62300
GEN62310
GEN62320
GEN62330
GEN62340
GEN62350
GEN62360
GEN62370
GEN62380
GEN62390
GEN62400
GEN62410
GEN62420
GEN62430
GEN62440
GEN62450
GEN62460
GEN62470
GEN62480
GEN62490
GEN62500
GEN62510
GEN62520
GEN62530
GEN62540
GEN62550
GEN62560
GEN62570
GEN62580
GEN62590
GEN62600
GEN62610
GEN62620
GEN62630
GEN62640
GEN62650
GEN62660
GEN62670
GEN62680
GEN62690
GEN62700
GEN62710
GEN62720
```

```
ELSE IF ((JJ.GT.3).AND.(CCC.GT.0.0)) THEN
    PHI=ABS(PHITWO(J)-PHIONE(J))
ENDIF
IF ((JJ.LT.3).AND.(CCC.EQ.0.0)) THEN
    PHI=ABS(PHIONE(J)-PHITWO(J-1))
ELSE IF ((JJ.GT.2).AND.(CCC.EQ.0.0)) THEN
    PHI=ABS(PHITWO(J)-PHIONE(J))
ENDIF
ENDIF
*****
*
*   SET THE NUMBER OF ELEMENTS ON AN EDGE.
*   THE NUMBER IS DEPENDENT UPON THE LENGTH OF THE
*   RIB OR CHANNEL
*
*****
IF (JJJ(J).EQ.2) THEN
    IF (PHI.GT.0.80) THEN
        NUMEA=3
    ELSE IF ((PHI.GT.0.40).AND.(PHI.LE.0.80)) THEN
        NUMEA=2
    ELSE
        NUMEA=2
    ENDIF
ELSE IF ((JJJ(J).EQ.1).OR.(JJJ(J).EQ.3)) THEN
    IF (PHI.GT.5.0) THEN
        NUMEA=3
    ELSE IF ((PHI.GT.3.0).AND.(PHI.LE.5.0)) THEN
        NUMEA=2
    ELSE
        NUMEA=2
    ENDIF
ENDIF
IF ((J.EQ.JSUR(ISUR)).AND.(JJ.EQ.1)) THEN
    NUMES(ISUR)=NUMEA
    ISUR=ISUR+1
```

```
GEN62730
GEN62740
GEN62750
GEN62760
GEN62770
GEN62780
GEN62790
GEN62800
GEN62810
GEN62820
GEN62830
GEN62840
GEN62850
GEN62860
GEN62870
GEN62880
GEN62890
GEN62900
GEN62910
GEN62920
GEN62930
GEN62940
GEN62950
GEN62960
GEN62970
GEN62980
GEN62990
GEN63000
GEN63010
GEN63020
GEN63030
GEN63040
GEN63050
GEN63060
GEN63070
GEN63080
GEN63090
GEN63100
GEN63110
GEN63120
GEN63130
GEN63140
GEN63150
GEN63160
GEN63170
GEN63180
GEN63190
GEN63200
GEN63210
GEN63220
GEN63230
GEN63240
GEN63250
GEN63260
GEN63270
GEN63280
GEN63290
GEN63300
GEN63310
GEN63320
GEN63330
GEN63340
GEN63350
GEN63360
```

ENDIF

IF (CCC.GT.0.0) THEN

IF ((JJJC(J).NE.0).AND.(JJ.LT.4)) GO TO 190

IF ((JJJC(J).NE.0).AND.(JJ.EQ.4)) I=I+3

IF ((JJJR(J).NE.0).AND.(JJ.GT.3)) GO TO 190

IF ((J.NE.1).AND.(JJJR(J-1).NE.0).AND.(JJ.EQ.1)) I=I+4

ELSE IF (CCC.EQ.0.0) THEN

IF ((JJJC(J).NE.0).AND.(JJ.LT.3)) GO TO 190

IF ((JJJC(J).NE.0).AND.(JJ.EQ.3)) I=I+3

IF ((JJJR(J).NE.0).AND.(JJ.GT.2)) GO TO 190

IF ((J.NE.1).AND.(JJJR(J-1).NE.0).AND.(JJ.EQ.1)) I=I+4

ENDIF

```
C *****
C *
C * BEGIN THE GENERATION OF THE COMMANDS THAT WILL *
C * CREATE THE MESHES OF THE INNER AND OUTER SHELLS *
C *
C *****
```

IA=I/100

IB=(I-IA\*100)/10

IC=I-IA\*100-IB\*10

IF (J.EQ.1) NUMEA=3

IF (IA.NE.0) THEN

```
65      WRITE(15,65) GENERL,IA,IB,IC
        FORMAT(1X,A5,'V',3I1)
```

ELSE IF (IB.NE.0) THEN

```
70      WRITE(15,70) GENERL,IB,IC
        FORMAT( 1X,A5,'V',2I1)
```

ELSE

```
75      WRITE(15,75) GENERL,IC
        FORMAT( 1X,A5,'V',I1)
```

ENDIF

MN=MN+1

IF (CCC.GT.0.0) THEN

IF ((JJ.EQ.2).OR.(JJ.EQ.5)) GO TO 85

```
80      WRITE(15,80) GENERL,NUMEA
        FORMAT(1X,A5,I1)
```

```
85      IF ((J.EQ.1).AND.(JJ.EQ.2)) THEN
```

GEN63370  
GEN63380  
GEN63390  
GEN63400  
GEN63410  
GEN63420  
GEN63430  
GEN63440  
GEN63450  
GEN63460  
GEN63470  
GEN63480  
GEN63490  
GEN63500  
GEN63510  
GEN63520  
GEN63530  
GEN63540  
GEN63550  
GEN63560  
GEN63570  
GEN63580  
GEN63590  
GEN63600  
GEN63610  
GEN63620  
GEN63630  
GEN63640  
GEN63650  
GEN63660  
GEN63670  
GEN63680  
GEN63690  
GEN63700  
GEN63710  
GEN63720  
GEN63730  
GEN63740  
GEN63750  
GEN63760  
GEN63770  
GEN63780  
GEN63790  
GEN63800  
GEN63810  
GEN63820  
GEN63830  
GEN63840  
GEN63850  
GEN63860  
GEN63870  
GEN63880  
GEN63890  
GEN63900  
GEN63910  
GEN63920  
GEN63930  
GEN63940  
GEN63950  
GEN63960  
GEN63970  
GEN63980  
GEN63990  
GEN64000

```
90      WRITE(15,90) GENERL
        FORMAT(1X,A5,'1')

        ENDIF

        IF ((JJJC(J).NE.0).AND.(JJ.EQ.5)) THEN

95          WRITE(15,95) GENERL
            FORMAT(1X,A5,'1')

            ENDIF

            IF ((JJJR(J-1).NE.0).AND.(JJ.EQ.2)) THEN

100              WRITE(15,100) GENERL
                FORMAT(1X,A5,'1')

                ENDIF

                WRITE(15,105) NOAL
105                FORMAT(2X,A8)

                IF ((J.EQ.1).AND.((JJ.EQ.1).OR.(JJ.EQ.3))) THEN

110                  WRITE(15,110) GENERL
                    FORMAT(1X,A5,'4')

                    WRITE(15,115) GENERL
115                    FORMAT(1X,A5,'1')

                    ENDIF

                    IF ((JJJC(J).NE.0).AND.((JJ.EQ.4).OR.(JJ.EQ.6))) THEN

                        IF (J.EQ.NUMBER) GO TO 130

120                          WRITE(15,120) GENERL
                            FORMAT(1X,A5,'4')

                            WRITE(15,125) GENERL
125                            FORMAT(1X,A5,'1')

                            ENDIF

                            IF (((JJJR(J-1).NE.0).AND.(J.NE.1)).AND.((JJ.EQ.1)
130                                .OR.(JJ.EQ.3))) THEN
                                #
                                WRITE(15,135) GENERL
135                                FORMAT(1X,A5,'4')

                                WRITE(15,140) GENERL
140                                FORMAT(1X,A5,'1')

                                ENDIF

                                ELSE IF (CCC.EQ.0.0) THEN

                                    WRITE(15,145) GENERL,NUMEA
145                                    FORMAT(1X,A5,I1)

                                    WRITE(15,150) NOAL
150                                    FORMAT(2X,A8)
```

```
GEN64010
GEN64020
GEN64030
GEN64040
GEN64050
GEN64060
GEN64070
GEN64080
GEN64090
GEN64100
GEN64110
GEN64120
GEN64130
GEN64140
GEN64150
GEN64160
GEN64170
GEN64180
GEN64190
GEN64200
GEN64210
GEN64220
GEN64230
GEN64240
GEN64250
GEN64260
GEN64270
GEN64280
GEN64290
GEN64300
GEN64310
GEN64320
GEN64330
GEN64340
GEN64350
GEN64360
GEN64370
GEN64380
GEN64390
GEN64400
GEN64410
GEN64420
GEN64430
GEN64440
GEN64450
GEN64460
GEN64470
GEN64480
GEN64490
GEN64500
GEN64510
GEN64520
GEN64530
GEN64540
GEN64550
GEN64560
GEN64570
GEN64580
GEN64590
GEN64600
GEN64610
GEN64620
GEN64630
GEN64640
```

```

      IF ((J.EQ.1).AND.((JJ.EQ.1).OR.(JJ.EQ.2))) THEN
      WRITE(15,155) GENERL
      FORMAT(1X,A5,'4')
      WRITE(15,160) GENERL
      FORMAT(1X,A5,'1')
      ENDIF
      IF ((JJJC(J).NE.0).AND.((JJ.EQ.3).OR.(JJ.EQ.4))) THEN
      IF (J.EQ.NUMBER) GO TO 175
      WRITE(15,165) GENERL
      FORMAT(1X,A5,'4')
      WRITE(15,170) GENERL
      FORMAT(1X,A5,'1')
      ENDIF
      IF (((JJJR(J-1).NE.0).AND.(J.NE.1)).AND.((JJ.EQ.1)
      .OR.(JJ.EQ.2))) THEN
      WRITE(15,180) GENERL
      FORMAT(1X,A5,'4')
      WRITE(15,185) GENERL
      FORMAT(1X,A5,'1')
      ENDIF
      ENDIF
      IF (CCC.GT.0.0) THEN
      IF (JJ.EQ.5) THEN
      I=I+2
      ELSE
      I=I+1
      ENDIF
      ELSE IF (CCC.EQ.0.0) THEN
      IF (JJ.EQ.1) THEN
      I=I+1
      ELSE
      I=I+2
      ENDIF
      ENDIF
      CONTINUE
      CONTINUE
      NTKMRC=MN
      WRITE(15,200)
      FORMAT(2X,'0 -1')

```



```
C *****
C *
C * BEGIN THE GENERATION OF THE COMMANDS THAT WILL
C * CREATE THE MESHES OF THE EXHAUST PORT
C *
C *****
C
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,54) GENERL
      WRITE(15,56) GENERL
      WRITE(15,58)
      I=NIVEP

      DO 240 JJ=1,IEND-1

          DO 235 IJJ=1,20

              IF (((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)).AND.
                  # (IJJ.EQ.13)) GO TO 240

              IA=I/100
              IB=(I-IA*100)/10
              IC=I-IA*100-IB*10

              IF (IA.NE.0) THEN

                  205 WRITE(15,205) GENERL,IA,IB,IC
                     FORMAT(1X,A5,'V',3I1)

                  ELSE IF (IB.NE.0) THEN

                  210 WRITE(15,210) GENERL,IB,IC
                     FORMAT( 1X,A5,'V',2I1)

                  ELSE

                  215 WRITE(15,215) GENERL,IC
                     FORMAT( 1X,A5,'V',I1)

                  ENDDIF

              MN=MN+1
              N=1
              IF (I.EQ.NIVEP) N=3

              IF ((JJ.EQ.1).OR.(IJJ.EQ.1).OR.(IJJ.EQ.13).OR.(IJJ.EQ.17)) THEN

                  IF ((JJ.EQ.6).AND.((IJJ.EQ.13).OR.(IJJ.EQ.17))) GO TO 230

                  IF ((IJJ.EQ.14).OR.(IJJ.EQ.15).OR.(IJJ.EQ.16).OR.
                  # (IJJ.EQ.18).OR.(IJJ.EQ.19).OR.(IJJ.EQ.20)) GO TO 230

                  DO 225 JJJJ=1,N

                      220 WRITE(15,220) GENERL
                         FORMAT(1X,A5,'1')

                      225 CONTINUE

                  ENDDIF

                  230 I=I+1
```

```
GEN65290
GEN65300
GEN65310
GEN65320
GEN65330
GEN65340
GEN65350
GEN65360
GEN65370
GEN65380
GEN65390
GEN65400
GEN65410
GEN65420
GEN65430
GEN65440
GEN65450
GEN65460
GEN65470
GEN65480
GEN65490
GEN65500
GEN65510
GEN65520
GEN65530
GEN65540
GEN65550
GEN65560
GEN65570
GEN65580
GEN65590
GEN65600
GEN65610
GEN65620
GEN65630
GEN65640
GEN65650
GEN65660
GEN65670
GEN65680
GEN65690
GEN65700
GEN65710
GEN65720
GEN65730
GEN65740
GEN65750
GEN65760
GEN65770
GEN65780
GEN65790
GEN65800
GEN65810
GEN65820
GEN65830
GEN65840
GEN65850
GEN65860
GEN65870
GEN65880
GEN65890
GEN65900
GEN65910
GEN65920
```

```
      IF (I.EQ.NIMVEP) GO TO 245
235      CONTINUE
240      CONTINUE
C      INTERMEDIATE EXHAUST *****
245      DO 280 L=1,12
          IA=I/100
          IB=(I-IA*100)/10
          IC=I-IA*100-IB*10
              IF (IA.NE.0) THEN
                  WRITE(15,250) GENERL, IA, IB, IC
250                  FORMAT(1X,A5,'V',3I1)
                  ELSE IF (IB.NE.0) THEN
                      WRITE(15,255) GENERL, IB, IC
255                      FORMAT( 1X,A5,'V',2I1)
                  ELSE
                      WRITE(15,260) GENERL, IC
260                      FORMAT( 1X,A5,'V',I1)
                  ENDIF
              MN=MN+1
              IF (L.EQ.1) THEN
                  N=3
              ELSE
                  N=1
              ENDIF
              IF ((L.EQ.6).OR.(L.EQ.7).OR.(L.EQ.8).OR.(L.EQ.10).OR.
#              (L.EQ.11).OR.(L.EQ.12)) GO TO 275
              DO 270 JJJJ=1,N
                  WRITE(15,265) GENERL
265                  FORMAT(1X,A5,'1')
270                  CONTINUE
275                  I=I+1
280                  CONTINUE
                  WRITE(15,200)
                  NTKMEX=MN-NTKMRC
C      *****
C      *
C      *      BEGIN THE GENERATION OF THE COMMANDS THAT WILL
C      *      CREATE THE MESHES OF THE INTAKE PORT
C      *
```

```
GEN65930
GEN65940
GEN65950
GEN65960
GEN65970
GEN65980
GEN65990
GEN66000
GEN66010
GEN66020
GEN66030
GEN66040
GEN66050
GEN66060
GEN66070
GEN66080
GEN66090
GEN66100
GEN66110
GEN66120
GEN66130
GEN66140
GEN66150
GEN66160
GEN66170
GEN66180
GEN66190
GEN66200
GEN66210
GEN66220
GEN66230
GEN66240
GEN66250
GEN66260
GEN66270
GEN66280
GEN66290
GEN66300
GEN66310
GEN66320
GEN66330
GEN66340
GEN66350
GEN66360
GEN66370
GEN66380
GEN66390
GEN66400
GEN66410
GEN66420
GEN66430
GEN66440
GEN66450
GEN66460
GEN66470
GEN66480
GEN66490
GEN66500
GEN66510
GEN66520
GEN66530
GEN66540
GEN66550
GEN66560
```

```
C *****
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,54) GENERL
      WRITE(15,56) GENERL
      WRITE(15,58)
      I=NIVIP
      DO 320 JJ=1,IEND-1
        DO 315 IJJ=1,24
          IF ((JJ.EQ.2).OR.(JJ.EQ.3).OR.(JJ.EQ.4)).AND.
#          (IJJ.EQ.15)) GO TO 320
          IA=I/100
          IB=(I-IA*100)/10
          IC=I-IA*100-IB*10
          IF (IA.NE.0) THEN
            WRITE(15,285) GENERL,IA,IB,IC
            FORMAT(1X,A5,'V',3I1)
          ELSE IF (IB.NE.0) THEN
            WRITE(15,290) GENERL,IB,IC
            FORMAT( 1X,A5,'V',2I1)
          ELSE
            WRITE(15,295) GENERL,IC
            FORMAT( 1X,A5,'V',I1)
          ENDIF
          MN=MN+1
          N=1
          IF (I.EQ.NIVIP) N=3
          IF ((JJ.EQ.1).OR.(IJJ.EQ.1).OR.(IJJ.EQ.15).OR.(IJJ.EQ.20)) THEN
            IF ((JJ.EQ.6).AND.((IJJ.EQ.15).OR.(IJJ.EQ.20))) GO TO 310
            IF ((IJJ.EQ.16).OR.(IJJ.EQ.17).OR.(IJJ.EQ.18).OR.
#            (IJJ.EQ.19).OR.(IJJ.EQ.21).OR.(IJJ.EQ.22).OR.
#            (IJJ.EQ.23).OR.(IJJ.EQ.24)) GO TO 310
            DO 305 JJJJ=1,N
              WRITE(15,300) GENERL
              FORMAT(1X,A5,'1')
            CONTINUE
          ENDIF
          I=I+1
          IF (I.EQ.NIMVIP) GO TO 325
          CONTINUE
        CONTINUE
      CONTINUE
```

GEN66570  
GEN66580  
GEN66590  
GEN66600  
GEN66610  
GEN66620  
GEN66630  
GEN66640  
GEN66650  
GEN66660  
GEN66670  
GEN66680  
GEN66690  
GEN66700  
GEN66710  
GEN66720  
GEN66730  
GEN66740  
GEN66750  
GEN66760  
GEN66770  
GEN66780  
GEN66790  
GEN66800  
GEN66810  
GEN66820  
GEN66830  
GEN66840  
GEN66850  
GEN66860  
GEN66870  
GEN66880  
GEN66890  
GEN66900  
GEN66910  
GEN66920  
GEN66930  
GEN66940  
GEN66950  
GEN66960  
GEN66970  
GEN66980  
GEN66990  
GEN67000  
GEN67010  
GEN67020  
GEN67030  
GEN67040  
GEN67050  
GEN67060  
GEN67070  
GEN67080  
GEN67090  
GEN67100  
GEN67110  
GEN67120  
GEN67130  
GEN67140  
GEN67150  
GEN67160  
GEN67170  
GEN67180  
GEN67190  
GEN67200

```
C      INTERMEDIATE INTAKE      *****
325      DO 350 L=1,ISTOP1
          IA=I/100
          IB=(I-IA*100)/10
          IC=I-IA*100-IB*10
          IF (IA.NE.0) THEN
              WRITE(15,330) GENERL,IA,IB,IC
              FORMAT(1X,A5,'V',3I1)
330          ELSE IF (IB.NE.0) THEN
              WRITE(15,335) GENERL,IB,IC
              FORMAT( 1X,A5,'V',2I1)
335          ELSE
              WRITE(15,340) GENERL,IC
              FORMAT( 1X,A5,'V',I1)
340          ENDIF
          MN=MN+1
          IF ((L.EQ.1).OR.(L.EQ.5).OR.(L.EQ.9).OR.(L.EQ.13)
              .OR.(L.EQ.17).OR.(L.EQ.21)) THEN
              WRITE(15,345) GENERL
              FORMAT(1X,A5,'1')
345          ENDIF
          I=I+1
350      CONTINUE
          NTKMIN=MN-NTKMRC-NTKMEX
          WRITE(15,200)
C      *****
C      *
C      *      BEGIN THE GENERATION OF THE COMMANDS THAT WILL
C      *      CREATE THE MESHES OF THE SPARK PLUG PORT
C      *
C      *****
          WRITE(15,30) SLASH
          WRITE(15,35) GENERL
          WRITE(15,40) GENERL
          WRITE(15,54) GENERL
          WRITE(15,56) GENERL
          WRITE(15,58)
          I=NIVSP1
          DO 405 IN=1,ISP
          DO 400 JJ=1,IEND-1
          DO 395 IJJ=1,20
```

```
GEN67210
GEN67220
GEN67230
GEN67240
GEN67250
GEN67260
GEN67270
GEN67280
GEN67290
GEN67300
GEN67310
GEN67320
GEN67330
GEN67340
GEN67350
GEN67360
GEN67370
GEN67380
GEN67390
GEN67400
GEN67410
GEN67420
GEN67430
GEN67440
GEN67450
GEN67460
GEN67470
GEN67480
GEN67490
GEN67500
GEN67510
GEN67520
GEN67530
GEN67540
GEN67550
GEN67560
GEN67570
GEN67580
GEN67590
GEN67600
GEN67610
GEN67620
GEN67630
GEN67640
GEN67650
GEN67660
GEN67670
GEN67680
GEN67690
GEN67700
GEN67710
GEN67720
GEN67730
GEN67740
GEN67750
GEN67760
GEN67770
GEN67780
GEN67790
GEN67800
GEN67810
GEN67820
GEN67830
GEN67840
```

```

IA=I/100
IB=(I-IA*100)/10
IC=I-IA*100-IB*10

      IF (IA.NE.0) THEN
355         WRITE(15,355) GENERL,IA,IB,IC
            FORMAT(1X,A5,'V',3I1)

      ELSE IF (IB.NE.0) THEN
360         WRITE(15,360) GENERL,IB,IC
            FORMAT( 1X,A5,'V',2I1)

      ELSE
365         WRITE(15,365) GENERL,IC
            FORMAT( 1X,A5,'V',I1)

      ENDIF

      MN=MN+1
      N=1
      IF ((I.EQ.NIVSP1).OR.(I.EQ.NIVSP2)) N=3

      IF (JJ.EQ.1) THEN

        IF ((IJJ.EQ.1).OR.(IJJ.EQ.2).OR.(IJJ.EQ.3).OR.(IJJ.EQ.4)
#         .OR.(IJJ.EQ.5).OR.(IJJ.EQ.6).OR.(IJJ.EQ.7).OR.(IJJ.EQ.8)
#         .OR.(IJJ.EQ.9).OR.(IJJ.EQ.17.) .OR.(IJJ.EQ.19)) THEN

          DO 375 JJJJ=1,N

370             WRITE(15,370) GENERL
                  FORMAT(1X,A5,'1')

375             CONTINUE

          ENDIF

          ELSE IF ((JJ.GT.1).AND.(IJJ.EQ.1)) THEN

            DO 385 JJJJ=1,N

380                WRITE(15,380) GENERL
                      FORMAT(1X,A5,'1')

385                CONTINUE

            ENDIF

            I=I+1

395                CONTINUE

400                CONTINUE

405 CONTINUE

      NTKMSP=MN-NTKMRC-NTKMEX-NTKMIN

C *****
C *
```

```

GEN67850
GEN67860
GEN67870
GEN67880
GEN67890
GEN67900
GEN67910
GEN67920
GEN67930
GEN67940
GEN67950
GEN67960
GEN67970
GEN67980
GEN67990
GEN68000
GEN68010
GEN68020
GEN68030
GEN68040
GEN68050
GEN68060
GEN68070
GEN68080
GEN68090
GEN68100
GEN68110
GEN68120
GEN68130
GEN68140
GEN68150
GEN68160
GEN68170
GEN68180
GEN68190
GEN68200
GEN68210
GEN68220
GEN68230
GEN68240
GEN68250
GEN68260
GEN68270
GEN68280
GEN68290
GEN68300
GEN68310
GEN68320
GEN68330
GEN68340
GEN68350
GEN68360
GEN68370
GEN68380
GEN68390
GEN68400
GEN68410
GEN68420
GEN68430
GEN68440
GEN68450
GEN68460
GEN68470
GEN68480
```

```
C *      BEGIN THE GENERATION OF THE COMMANDS THAT WILL *      GEN68490
C *      CREATE THE MESHES OF THE RIBS *      GEN68500
C *      *      *      GEN68510
C ***** *      GEN68520
C ***** *      GEN68530
C ***** *      GEN68540
C *      *      GEN68550
C *      SET ELEMENT TYPE FOR GENERATION OF SOLID ELEMENTS *      GEN68560
C *      *      GEN68570
C ***** *      GEN68580
C ***** *      GEN68590
C ***** *      GEN68600
C ***** *      GEN68610
C ***** *      GEN68620
C ***** *      GEN68630
C ***** *      GEN68640
C ***** *      GEN68650
C ***** *      GEN68660
C ***** *      GEN68670
C ***** *      GEN68680
C ***** *      GEN68690
C ***** *      GEN68700
C ***** *      GEN68710
C ***** *      GEN68720
C ***** *      GEN68730
C ***** *      GEN68740
C ***** *      GEN68750
C ***** *      GEN68760
C ***** *      GEN68770
C ***** *      GEN68780
C ***** *      GEN68790
C ***** *      GEN68800
C ***** *      GEN68810
C ***** *      GEN68820
C ***** *      GEN68830
C ***** *      GEN68840
C ***** *      GEN68850
C ***** *      GEN68860
C ***** *      GEN68870
C ***** *      GEN68880
C ***** *      GEN68890
C ***** *      GEN68900
C ***** *      GEN68910
C ***** *      GEN68920
C ***** *      GEN68930
C ***** *      GEN68940
C ***** *      GEN68950
C ***** *      GEN68960
C ***** *      GEN68970
C ***** *      GEN68980
C ***** *      GEN68990
C ***** *      GEN69000
C ***** *      GEN69010
C ***** *      GEN69020
C ***** *      GEN69030
C ***** *      GEN69040
C ***** *      GEN69050
C ***** *      GEN69060
C ***** *      GEN69070
C ***** *      GEN69080
C ***** *      GEN69090
C ***** *      GEN69100
C ***** *      GEN69110
C ***** *      GEN69120

      WRITE(15,200)

      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL

      WRITE(15,420) GENERL
420      FORMAT(1X,A5,'EL')

      WRITE(15,425) GENERL
425      FORMAT(1X,A5,'DEF')

      WRITE(15,430) GENERL
430      FORMAT(1X,A5,'TY')

      WRITE(15,435) GENERL
435      FORMAT(1X,A5,'SOL')

      WRITE(15,440) GENERL
440      FORMAT(1X,A5,'PB')

      WRITE(15,30) SLASH
      WRITE(15,54) GENERL
      WRITE(15,56) GENERL
      WRITE(15,58)

      I=6
      NUMEA=3

      DO 505 J=1,NUMBER

        IF (JJJR(J).NE.0) GO TO 500

        IA=I/100
        IB=(I-IA*100)/10
        IC=I-IA*100-IB*10

        IF (IA.NE.0) THEN

          WRITE(15,465) GENERL,IA,IB,IC
465          FORMAT(1X,A5,'V',3I1)

          ELSE IF (IB.NE.0) THEN

            WRITE(15,470) GENERL,IB,IC
470            FORMAT(1X,A5,'V',2I1)

            ELSE

              WRITE(15,475) GENERL,IC
475              FORMAT(1X,A5,'V',I1)

            ENDIF
```

```

      MN=MN+1
      IF (IRIB(J).EQ.1) THEN
        WRITE(15,480) GENERL,NUMEA
480      FORMAT(1X,A5,I1)
        WRITE(15,485) NOAL
485      FORMAT(2X,A8)
      ELSE
        WRITE(15,490) NOAL
490      FORMAT(2X,A8)
        WRITE(15,495) GENERL,NUMEA
495      FORMAT(1X,A5,I1)
      ENDIF
500      I=I+7
505      CONTINUE

      NSLMRB=MN-NTKMRC-NTKMEX-NTKMIN-NTKMSP

C *****
C *
C * BEGIN THE GENERATION OF THE COMMANDS THAT WILL
C * CREATE THE MESHES OF THE STIFFENED CHANNELS
C *
C *****
C *****
C *
C * SET ELEMENT TYPE FOR GENERATION OF THIN SHELLS
C *
C *****

      WRITE(15,200)
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL

      WRITE(15,420) GENERL
      WRITE(15,425) GENERL
      WRITE(15,430) GENERL

      WRITE(15,520) GENERL
520      FORMAT(1X,A5,'TN')

      WRITE(15,525) GENERL
525      FORMAT(1X,A5,'PQ')

      WRITE(15,30) SLASH
      WRITE(15,54) GENERL
      WRITE(15,56) GENERL
      WRITE(15,58)

      DO 575 JJ=1,NSTCN+1

        I=JJJSC(JJ)
        IA=I/1000
```

```

GEN69130
GEN69140
GEN69150
GEN69160
GEN69170
GEN69180
GEN69190
GEN69200
GEN69210
GEN69220
GEN69230
GEN69240
GEN69250
GEN69260
GEN69270
GEN69280
GEN69290
GEN69300
GEN69310
GEN69320
GEN69330
GEN69340
GEN69350
GEN69360
GEN69370
GEN69380
GEN69390
GEN69400
GEN69410
GEN69420
GEN69430
GEN69440
GEN69450
GEN69460
GEN69470
GEN69480
GEN69490
GEN69500
GEN69510
GEN69520
GEN69530
GEN69540
GEN69550
GEN69560
GEN69570
GEN69580
GEN69590
GEN69600
GEN69610
GEN69620
GEN69630
GEN69640
GEN69650
GEN69660
GEN69670
GEN69680
GEN69690
GEN69700
GEN69710
GEN69720
GEN69730
GEN69740
GEN69750
GEN69760
```

```
IB=(I-IA*1000)/100
IC=(I-IA*1000-IB*100)/10
ID=I-IA*1000-IB*100-IC*10
```

```
IF (IA.NE.0) THEN
```

545

```
WRITE(15,545) GENERL,IA,IB,IC,ID
FORMAT(1X,A5,'SU',4I1)
```

```
ELSE IF (IB.NE.0) THEN
```

550

```
WRITE(15,550) GENERL,IB,IC,ID
FORMAT(1X,A5,'SU',3I1)
```

```
ELSEIF (IC.NE.0) THEN
```

555

```
WRITE(15,555) GENERL,IC,ID
FORMAT(1X,A5,'SU',2I1)
```

```
ELSE
```

560

```
WRITE(15,560) GENERL,ID
FORMAT(1X,A5,'SU',I1)
```

```
ENDIF
```

```
MN=MN+1
```

565

```
WRITE(15,565) GENERL,NUMS(JJ)
FORMAT(1X,A5,I1)
```

```
WRITE(15,485) NOAL
```

570

```
WRITE(15,570) GENERL
FORMAT(1X,A5,'3')
```

575 CONTINUE

```
NTNMSC=MN-NTKMRC-NTKMEX-NTKMIN-NTKMSP-NSLMRB
```

```
C *****
C *
C *
C *
C *
C *****
```

```
END MESH GENERATION
```

```
C *****
C *
C *
C *
C *
C *
C *****
```

```
BEGIN THE GENERATION OF THE COMMANDS THAT WILL
CREATE THE NODES AND ELEMENTS
```

```
C *****
C *
C *
C *
C *
C *
C *****
```

```
BEGIN NODE AND ELEMENT GENERATION
OF THE INNER AND OUTER SHELLS
```

```
C *****
C *
C *
C *
C *****
```

```
SET THE TWO DIFFERENT MATERIAL PROPERTIES
```

GEN69770  
GEN69780  
GEN69790  
GEN69800  
GEN69810  
GEN69820  
GEN69830  
GEN69840  
GEN69850  
GEN69860  
GEN69870  
GEN69880  
GEN69890  
GEN69900  
GEN69910  
GEN69920  
GEN69930  
GEN69940  
GEN69950  
GEN69960  
GEN69970  
GEN69980  
GEN69990  
GEN70000  
GEN70010  
GEN70020  
GEN70030  
GEN70040  
GEN70050  
GEN70060  
GEN70070  
GEN70080  
GEN70090  
GEN70100  
GEN70110  
GEN70120  
GEN70130  
GEN70140  
GEN70150  
GEN70160  
GEN70170  
GEN70180  
GEN70190  
GEN70200  
GEN70210  
GEN70220  
GEN70230  
GEN70240  
GEN70250  
GEN70260  
GEN70270  
GEN70280  
GEN70290  
GEN70300  
GEN70310  
GEN70320  
GEN70330  
GEN70340  
GEN70350  
GEN70360  
GEN70370  
GEN70380  
GEN70390  
GEN70400



```
C *****
      WRITE(15,200)
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,420) GENERL
      WRITE(15,580) GENERL
580  FORMAT(1X,A5,'MAT')
      WRITE(15,585) GENERL
585  FORMAT(1X,A5,'EN')
      WRITE(15,590) GENERL
590  FORMAT(1X,A5,'1')
      WRITE(15,595) GENERL
595  FORMAT(1X,A5,'10E6')
      WRITE(15,600) GENERL
600  FORMAT(1X,A5,'0.30')
      DO 610 J=1,9
      WRITE(15,605)
605  FORMAT(2X,'0 -1')
610  CONTINUE
      WRITE(15,585) GENERL
      WRITE(15,615) GENERL
615  FORMAT(1X,A5,'2')
      WRITE(15,620) GENERL
620  FORMAT(1X,A5,'20E6')
      WRITE(15,625) GENERL
625  FORMAT(1X,A5,'0.25')
      DO 630 J=1,9
      WRITE(15,605)
630  CONTINUE
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,30) SLASH
      WRITE(15,420) GENERL
      WRITE(15,425) GENERL
      WRITE(15,430) GENERL
      WRITE(15,48) GENERL
      WRITE(15,50) GENERL
      WRITE(15,30) SLASH
C *****
C *
C *
C *
C *
C *****
      END MATERIAL PROPERTY SET
C *****
```

```
GEN70410
GEN70420
GEN70430
GEN70440
GEN70450
GEN70460
GEN70470
GEN70480
GEN70490
GEN70500
GEN70510
GEN70520
GEN70530
GEN70540
GEN70550
GEN70560
GEN70570
GEN70580
GEN70590
GEN70600
GEN70610
GEN70620
GEN70630
GEN70640
GEN70650
GEN70660
GEN70670
GEN70680
GEN70690
GEN70700
GEN70710
GEN70720
GEN70730
GEN70740
GEN70750
GEN70760
GEN70770
GEN70780
GEN70790
GEN70800
GEN70810
GEN70820
GEN70830
GEN70840
GEN70850
GEN70860
GEN70870
GEN70880
GEN70890
GEN70900
GEN70910
GEN70920
GEN70930
GEN70940
GEN70950
GEN70960
GEN70970
GEN70980
GEN70990
GEN71000
GEN71010
GEN71020
GEN71030
GEN71040
```

```
635      WRITE(15,635) GENERL
      FORMAT(1X,A5,'NE')

      WRITE(15,58)

640      WRITE(15,640) GENERL,NTKMRC
      FORMAT(1X,A5,'1',1X,I3,1X,'1')

      I=0

      DO 665 JJ=1,NUMBER

          DO 660 IJJ=1,ISTOP

              I=I+1
              IF (I.GT.NTKMRC) GO TO 670

              WRITE(15,645) GENERL
              FORMAT(1X,A5,'1')

                  IF ((JJ.EQ.1).AND.(IJJ.EQ.1)) THEN

                      WRITE(15,650) YES
                      FORMAT(2X,A8)

                      ENDIF

                      IF ((CCC.GT.0.0).AND.((IJJ.EQ.1).OR.(IJJ.EQ.4))) THEN

                          WRITE(15,655) GENERL
                          FORMAT(1X,A5,'2')

                          ELSE

                              WRITE(15,645) GENERL

                              ENDIF

                              IF ((JJ.EQ.1).AND.(IJJ.EQ.1)) THEN

                                  WRITE(15,650) YES

                                  ENDIF

660          CONTINUE
665      CONTINUE

C *****
C *
C *          BEGIN NODE AND ELEMENT GENERATION
C *          OF THE EXHAUST PORT
C *
C *****

670      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,635) GENERL
      WRITE(15,58)

      WRITE(15,675) GENERL,NTKMRC+1,NTKMRC+NTKMEX
      FORMAT(1X,A5,I3,1X,I3,1X,'1')

      DO 680 JJ=1,NTKMEX
```

```

WRITE(15,645) GENERL
      IF (JJ.EQ.1) THEN
        WRITE(15,650) YES
      ENDIF
      IF ((CCC.GT.0.0).AND.((IJJ.EQ.1).OR.(IJJ.EQ.4))) THEN
        WRITE(15,655) GENERL
      ELSE
        WRITE(15,645) GENERL
      ENDIF
      IF (JJ.EQ.1) THEN
        WRITE(15,650) YES
      ENDIF
680      CONTINUE
      WRITE(15,30) SLASH
      C *****
      C *
      C *          BEGIN NODE AND ELEMENT GENERATION
      C *          OF THE INTAKE PORT
      C *
      C *****
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      NTW1=1
      DO 695 JJ=1,ISTOP3
        WRITE(15,635) GENERL
        WRITE(15,58)
        IF ((JJ.EQ.1).OR.(JJ.EQ.12)) THEN
          WRITE(15,685) GENERL,NTKMRC+NTKMEX+NTW1
685      FORMAT(1X,A5,I3)
          NTW2=NTW1
        ELSE
          WRITE(15,675) GENERL,NTKMRC+NTKMEX+NTW1,NTKMRC+NTKMEX+NTW2
        ENDIF
        III=1
      IF ((JJ.NE.1).OR.(JJ.NE.12)) THEN
        DO 690 K=NTKMRC+NTKMEX+NTW1,NTKMRC+NTKMEX+NTW2

```

```

GEN71690
GEN71700
GEN71710
GEN71720
GEN71730
GEN71740
GEN71750
GEN71760
GEN71770
GEN71780
GEN71790
GEN71800
GEN71810
GEN71820
GEN71830
GEN71840
GEN71850
GEN71860
GEN71870
GEN71880
GEN71890
GEN71900
GEN71910
GEN71920
GEN71930
GEN71940
GEN71950
GEN71960
GEN71970
GEN71980
GEN71990
GEN72000
GEN72010
GEN72020
GEN72030
GEN72040
GEN72050
GEN72060
GEN72070
GEN72080
GEN72090
GEN72100
GEN72110
GEN72120
GEN72130
GEN72140
GEN72150
GEN72160
GEN72170
GEN72180
GEN72190
GEN72200
GEN72210
GEN72220
GEN72230
GEN72240
GEN72250
GEN72260
GEN72270
GEN72280
GEN72290
GEN72300
GEN72310
GEN72320

```

```
        WRITE(15,645) GENERL
                                IF (III.EQ.1) THEN
                                    WRITE(15,650) YES
                                ENDIF
        IF ((CCC.GT.0.0).AND.((IJJ.EQ.1).OR.(IJJ.EQ.4))) THEN
            WRITE(15,655) GENERL
        ELSE
            WRITE(15,645) GENERL
        ENDIF
                                IF (III.EQ.1) THEN
                                    WRITE(15,650) YES
                                ENDIF
        III=III+1
690      CONTINUE
        ELSE IF ((JJ.EQ.1).OR.(JJ.EQ.12)) THEN
            WRITE(15,645) GENERL
            WRITE(15,645) GENERL
            WRITE(15,650) YES
        ENDIF
        IF (JJ.EQ.1) NTW1=3
        IF (JJ.EQ.1) NTW2=12
        IF (JJ.EQ.2) NTW1=14
        IF (JJ.EQ.2) NTW2=25
        IF (JJ.EQ.3) NTW1=27
        IF (JJ.EQ.3) NTW2=36
        IF (JJ.EQ.4) NTW1=38
        IF (JJ.EQ.4) NTW2=39
        IF (JJ.EQ.5) NTW1=41
        IF (JJ.EQ.5) NTW2=50
        IF (JJ.EQ.6) NTW1=52
        IF (JJ.EQ.6) NTW2=53
        IF (JJ.EQ.7) NTW1=55
        IF (JJ.EQ.7) NTW2=64
        IF (JJ.EQ.8) NTW1=66
        IF (JJ.EQ.8) NTW2=67
        IF (JJ.EQ.9) NTW1=69
        IF (JJ.EQ.9) NTW2=78
        IF (JJ.EQ.10) NTW1=80
        IF (JJ.EQ.10) NTW2=90
        IF ((CCC.EQ.0.0).AND.(JJ.EQ.10)) NTW2=NTW2+24
        IF (JJ.EQ.11) NTW1=91
        IF (JJ.EQ.12) NTW1=93
        IF (JJ.EQ.12) NTW2=102
        IF (JJ.EQ.13) NTW1=104
        IF (JJ.EQ.13) NTW2=114+36
695      CONTINUE
```

```
GEN72330
GEN72340
GEN72350
GEN72360
GEN72370
GEN72380
GEN72390
GEN72400
GEN72410
GEN72420
GEN72430
GEN72440
GEN72450
GEN72460
GEN72470
GEN72480
GEN72490
GEN72500
GEN72510
GEN72520
GEN72530
GEN72540
GEN72550
GEN72560
GEN72570
GEN72580
GEN72590
GEN72600
GEN72610
GEN72620
GEN72630
GEN72640
GEN72650
GEN72660
GEN72670
GEN72680
GEN72690
GEN72700
GEN72710
GEN72720
GEN72730
GEN72740
GEN72750
GEN72760
GEN72770
GEN72780
GEN72790
GEN72800
GEN72810
GEN72820
GEN72830
GEN72840
GEN72850
GEN72860
GEN72870
GEN72880
GEN72890
GEN72900
GEN72910
GEN72920
GEN72930
GEN72940
GEN72950
GEN72960
```

```
WRITE(15,30) SLASH
C *****
C *
C * BEGIN NODE AND ELEMENT GENERATION
C * OF THE SPARK PLUG PORT
C *
C *****

WRITE(15,30) SLASH
WRITE(15,35) GENERL
WRITE(15,40) GENERL
WRITE(15,30) SLASH
WRITE(15,420) GENERL
WRITE(15,425) GENERL
WRITE(15,430) GENERL
WRITE(15,48) GENERL
WRITE(15,50) GENERL
WRITE(15,30) SLASH

WRITE(15,635) GENERL
WRITE(15,58)

WRITE(15,675) GENERL,NTKMRC+NTKMEX+NTKMIN+1
# ,NTKMRC+NTKMEX+NTKMIN+NTKMSP

DO 700 JJ=1,NTKMSP

WRITE(15,645) GENERL

IF (JJ.EQ.1) THEN

WRITE(15,650) YES

ENDIF

IF ((CCC.GT.0.0).AND.((IJJ.EQ.1).OR.(IJJ.EQ.4))) THEN

WRITE(15,655) GENERL

ELSE

WRITE(15,645) GENERL

ENDIF

IF (JJ.EQ.1) THEN

WRITE(15,650) YES

ENDIF

700 CONTINUE

WRITE(15,30) SLASH
C *****
C *
C * BEGIN NODE AND ELEMENT GENERATION
C * TRIANGULAR WEDGES IN THE INTAKE PORT
C *
C *****
C *****
```

GEN72970  
GEN72980  
GEN72990  
GEN73000  
GEN73010  
GEN73020  
GEN73030  
GEN73040  
GEN73050  
GEN73060  
GEN73070  
GEN73080  
GEN73090  
GEN73100  
GEN73110  
GEN73120  
GEN73130  
GEN73140  
GEN73150  
GEN73160  
GEN73170  
GEN73180  
GEN73190  
GEN73200  
GEN73210  
GEN73220  
GEN73230  
GEN73240  
GEN73250  
GEN73260  
GEN73270  
GEN73280  
GEN73290  
GEN73300  
GEN73310  
GEN73320  
GEN73330  
GEN73340  
GEN73350  
GEN73360  
GEN73370  
GEN73380  
GEN73390  
GEN73400  
GEN73410  
GEN73420  
GEN73430  
GEN73440  
GEN73450  
GEN73460  
GEN73470  
GEN73480  
GEN73490  
GEN73500  
GEN73510  
GEN73520  
GEN73530  
GEN73540  
GEN73550  
GEN73560  
GEN73570  
GEN73580  
GEN73590  
GEN73600

```
C  *
C  *      SET THE ELEMENT TYPE FOR PARABOLIC WEDGES      *
C  *
C  *****
C
      WRITE(15,30) SLASH
      WRITE(15,35) GENERL
      WRITE(15,40) GENERL
      WRITE(15,420) GENERL
      WRITE(15,425) GENERL
      WRITE(15,430) GENERL
      WRITE(15,705) GENERL
705  FORMAT(1X,A5,'TK')
      WRITE(15,710) GENERL
710  FORMAT(1X,A5,'PB')
      WRITE(15,30) SLASH
      DO 720 JJ=1,ISTOP2
      WRITE(15,635) GENERL
      WRITE(15,58)
      WRITE(15,715) GENERL,NTKMRC+NTKMEX+NTW
715  FORMAT(1X,A5,I3)
      WRITE(15,645) GENERL
      WRITE(15,650) YES
      IF ((CCC.GT.0.0).AND.((IJJ.EQ.1).OR.(IJJ.EQ.4))) THEN
      WRITE(15,655) GENERL
      ELSE
      WRITE(15,645) GENERL
      ENDIF
      WRITE(15,650) YES
      IF (JJ.EQ.1) NTW=13
      IF (JJ.EQ.2) NTW=26
      IF (JJ.EQ.3) NTW=37
      IF (JJ.EQ.4) NTW=40
      IF (JJ.EQ.5) NTW=51
      IF (JJ.EQ.6) NTW=54
      IF (JJ.EQ.7) NTW=65
      IF (JJ.EQ.8) NTW=68
      IF (JJ.EQ.9) NTW=79
      IF (JJ.EQ.10) NTW=92
      IF (JJ.EQ.11) NTW=103
720  CONTINUE
      WRITE(15,30) SLASH
C  *****
C  *
C  *      BEGIN NODE AND ELEMENT GENERATION      *
C  *
```

```
GEN73610
GEN73620
GEN73630
GEN73640
GEN73650
GEN73660
GEN73670
GEN73680
GEN73690
GEN73700
GEN73710
GEN73720
GEN73730
GEN73740
GEN73750
GEN73760
GEN73770
GEN73780
GEN73790
GEN73800
GEN73810
GEN73820
GEN73830
GEN73840
GEN73850
GEN73860
GEN73870
GEN73880
GEN73890
GEN73900
GEN73910
GEN73920
GEN73930
GEN73940
GEN73950
GEN73960
GEN73970
GEN73980
GEN73990
GEN74000
GEN74010
GEN74020
GEN74030
GEN74040
GEN74050
GEN74060
GEN74070
GEN74080
GEN74090
GEN74100
GEN74110
GEN74120
GEN74130
GEN74140
GEN74150
GEN74160
GEN74170
GEN74180
GEN74190
GEN74200
GEN74210
GEN74220
GEN74230
GEN74240
```

```
C * FOR THE RIBS * GEN74250
C * * GEN74260
C ***** GEN74270
C ***** GEN74280
C ***** GEN74290
C * * GEN74300
C * SET THE ELEMENT TYPE FOR SOLID ELEMENTS * GEN74310
C * * GEN74320
C ***** GEN74330
C ***** GEN74340
C ***** GEN74350
C ***** GEN74360
C ***** GEN74370
C ***** GEN74380
C ***** GEN74390
C ***** GEN74400
C ***** GEN74410
C ***** GEN74420
C ***** GEN74430
C ***** GEN74440
C ***** GEN74450
C ***** GEN74460
C ***** GEN74470
C ***** GEN74480
C ***** GEN74490
C * * GEN74500
C * SET THE PHYSICAL PROPERTY FOR THIN SHELLS * GEN74510
C * * GEN74520
C ***** GEN74530
C ***** GEN74540
C ***** GEN74550
C ***** GEN74560
C ***** GEN74570
C ***** GEN74580
C ***** GEN74590
C ***** GEN74600
C ***** GEN74610
C ***** GEN74620
C ***** GEN74630
C ***** GEN74640
C ***** GEN74650
C ***** GEN74660
C ***** GEN74670
C ***** GEN74680
C ***** GEN74690
C ***** GEN74700
C ***** GEN74710
C ***** GEN74720
C ***** GEN74730
C ***** GEN74740
C ***** GEN74750
C ***** GEN74760
C ***** GEN74770
C ***** GEN74780
C ***** GEN74790
C ***** GEN74800
C ***** GEN74810
C ***** GEN74820
C ***** GEN74830
C ***** GEN74840
C ***** GEN74850
C ***** GEN74860
C ***** GEN74870
C ***** GEN74880

WRITE(15,30) SLASH
WRITE(15,35) GENERL
WRITE(15,40) GENERL
WRITE(15,420) GENERL
WRITE(15,425) GENERL
WRITE(15,430) GENERL

WRITE(15,725) GENERL
FORMAT(1X,A5,'SOL')

WRITE(15,50) GENERL
WRITE(15,30) SLASH
WRITE(15,420) GENERL

C *****
C *
C * SET THE PHYSICAL PROPERTY FOR THIN SHELLS *
C * *
C *****

800 WRITE(15,800) GENERL
FORMAT(1X,A5,'PHY')

805 WRITE(15,805) GENERL
FORMAT(1X,A5,'EN')

810 WRITE(15,810) GENERL
FORMAT(1X,A5,'3')

815 WRITE(15,815) GENERL
FORMAT(1X,A5)

WRITE(15,815) GENERL
WRITE(15,815) GENERL
WRITE(15,815) GENERL

WRITE(15,30) SLASH
WRITE(15,635) GENERL
WRITE(15,58)

WRITE(15,675) GENERL,NTKMRC+NTKMEX+NTKMIN+NTKMSP+1
# ,NTKMRC+NTKMEX+NTKMIN+NTKMSP+NSLMRB

DO 820 JJ=1,NSLMRB

WRITE(15,810) GENERL

WRITE(15,645) GENERL

IF (JJ.EQ.1) THEN

WRITE(15,650) YES
```

ENDIF

```
820          CONTINUE

          WRITE(15,30) SLASH

C *****
C *
C *          BEGIN NODE AND ELEMENT GENERATION
C *          FOR THE STIFFENED CHANNELS
C *
C *****

C *****
C *
C *          SET THE ELEMENT TYPE FOR THIN SHELLS
C *
C *****

          WRITE(15,30) SLASH
          WRITE(15,35) GENERL
          WRITE(15,40) GENERL
          WRITE(15,420) GENERL
          WRITE(15,425) GENERL
          WRITE(15,430) GENERL

          WRITE(15,825) GENERL
825          FORMAT(1X,A5,'TN')

          WRITE(15,830) GENERL
830          FORMAT(1X,A5,'PQ')

          WRITE(15,30) SLASH
          WRITE(15,420) GENERL

C *****
C *
C *          SET THE PHYSICAL PROPERTY FOR THIN SHELLS
C *
C *****

          WRITE(15,835) GENERL
835          FORMAT(1X,A5,'PHY')

          WRITE(15,840) GENERL
840          FORMAT(1X,A5,'EN')

          WRITE(15,845) GENERL
845          FORMAT(1X,A5,'2')

          WRITE(15,850) GENERL
850          FORMAT(1X,A5,'0.25,0,0,0')

          WRITE(15,855) GENERL
855          FORMAT(1X,A5)

          WRITE(15,855) GENERL
          WRITE(15,855) GENERL
          WRITE(15,855) GENERL
          WRITE(15,855) GENERL
          WRITE(15,855) GENERL
          WRITE(15,855) GENERL

          WRITE(15,30) SLASH
```

```
GEN74890
GEN74900
GEN74910
GEN74920
GEN74930
GEN74940
GEN74950
GEN74960
GEN74970
GEN74980
GEN74990
GEN75000
GEN75010
GEN75020
GEN75030
GEN75040
GEN75050
GEN75060
GEN75070
GEN75080
GEN75090
GEN75100
GEN75110
GEN75120
GEN75130
GEN75140
GEN75150
GEN75160
GEN75170
GEN75180
GEN75190
GEN75200
GEN75210
GEN75220
GEN75230
GEN75240
GEN75250
GEN75260
GEN75270
GEN75280
GEN75290
GEN75300
GEN75310
GEN75320
GEN75330
GEN75340
GEN75350
GEN75360
GEN75370
GEN75380
GEN75390
GEN75400
GEN75410
GEN75420
GEN75430
GEN75440
GEN75450
GEN75460
GEN75470
GEN75480
GEN75490
GEN75500
GEN75510
GEN75520
```



```
WRITE(15,635) GENERL
WRITE(15,58)
```

```
WRITE(15,675) GENERL,NTKMRC+NTKMEX+NTKMIN+NTKMSP+NSLMRB+1
# ,NTKMRC+NTKMEX+NTKMIN+NTKMSP+NSLMRB+NTNMSC
```

```
DO 865 JJ=1,NTNMSC
```

```
WRITE(15,860) GENERL
FORMAT(1X,A5,'2')
```

```
WRITE(15,645) GENERL
```

```
IF (JJ.EQ.1) THEN
```

```
WRITE(15,650) YES
```

```
ENDIF
```

```
865 CONTINUE
```

```
WRITE(15,30) SLASH
```

```
PRINT*, ' '
PRINT*, ' THE UNIVERSAL AND PROGRAM FILES HAVE BEEN CREATED.'
PRINT*, ' '
PRINT*, ' BEFORE THE UNIVERSAL FILE CAN BE READ AND THE'
PRINT*, ' PROGRAM FILE EXECUTED, THE USER MUST ENTER THE'
PRINT*, ' SDRC SOFTWARE.'
```

```
PRINT*, ' '
PRINT*, ' NOTE THAT A FILE CALLED "MODEL DATA" CAN NOT'
PRINT*, ' EXIST ON YOUR DISK. IF IT DOES, AN ERROR WILL'
PRINT*, ' RESULT WHEN YOU ENTER THE SDRC SOFTWARE. TO'
PRINT*, ' CORRECT THE ERROR, CHANGE THE NAME OF THE'
PRINT*, ' "MODEL DATA" FILE THAT EXISTS ON YOUR DISK'
PRINT*, ' TO A DIFFERENT NAME.'
PRINT*, ' '
PRINT*, ' AFTER ENTERING THE SDRC SOFTWARE IN THE PROGRAM MODE,'
PRINT*, ' RESPOND "R" (FOR RUN) TO THE FIRST QUESTION,'
PRINT*, ' RESPOND "GO" TO THE SECOND QUESTION AND'
PRINT*, ' IN RESPONSE TO THE THIRD QUESTION,'
PRINT*, ' ENTER THE TERMINAL TYPE THAT YOU ARE USING.'
```

```
C *****
C *
C * END NODE AND ELEMENT GENERATION
C *
C *****
```

```
RETURN
END
```

```
C *****
C *
C * THIS PROGRAM GENERATES A PROGRAM FILE THAT, IN
C * TURN CALLS A PROGRAM FILE. THE PROGRAM FILE
C * FILE GENERATED IS CALLED RUN. THE RUN PROGRAM FILE
C * READS THE UNIVERSAL FILE - GEOMETRY- AND RUNS THE
C * PROGRAM FILE - MSNDEL.
C *
C *****
```

GEN75530  
GEN75540  
GEN75550  
GEN75560  
GEN75570  
GEN75580  
GEN75590  
GEN75600  
GEN75610  
GEN75620  
GEN75630  
GEN75640  
GEN75650  
GEN75660  
GEN75670  
GEN75680  
GEN75690  
GEN75700  
GEN75710  
GEN75720  
GEN75730  
GEN75740  
GEN75750  
GEN75760  
GEN75770  
GEN75780  
GEN75790  
GEN75800  
GEN75810  
GEN75820  
GEN75830  
GEN75840  
GEN75850  
GEN75860  
GEN75870  
GEN75880  
GEN75890  
GEN75900  
GEN75910  
GEN75920  
GEN75930  
GEN75940  
GEN75950  
GEN75960  
GEN75970  
GEN75980  
GEN75990  
GEN76000  
GEN76010  
GEN76020  
GEN76030  
GEN76040  
GEN76050  
GEN76060  
GEN76070  
GEN76080  
GEN76090  
GEN76100  
GEN76110  
GEN76120  
GEN76130  
GEN76140  
GEN76150  
GEN76160

```
SUBROUTINE GO (ICONTI)
CHARACTER*10 GENERL,YES
    GENERL='10 -1'
    YES='3 -1YES'
    WRITE(22,25)
    25    FORMAT(2X,'4 -1&L')
    WRITE(22,30)
    30    FORMAT(2X,'5 -1MODEL')
    WRITE(22,35) YES
    35    FORMAT(2X,A8)
    WRITE(22,40)
    40    FORMAT(1X,'65 -1FINITE ELEMENT MODEL OF A ROTARY COMBUSTION
#ENGINE CENTER HOUSING')
    WRITE(22,45)
    45    FORMAT(2X,'2 -1IN')
    WRITE(22,50)
    50    FORMAT(2X,'3 -1FEM')
    WRITE(22,55)
    55    FORMAT(2X,'2 -1FT')
    WRITE(22,60)
    60    FORMAT(2X,'2 -1RF')
    WRITE(22,65)
    65    FORMAT(2X,'2 -1UN')
    WRITE(22,70)
    70    FORMAT(2X,'8 -1GEOMETRY')
    WRITE(22,71)
    71    FORMAT(2X,'2 -1AU')
    WRITE(22,72)
    72    FORMAT(1X,'12 -1/ DO E V OFF')
    WRITE(22,73)
    73    FORMAT(2X,'6 -1SU OFF')
    WRITE(22,74)
    74    FORMAT(2X,'2 -1DR')
    IF (ICONTI.EQ.2) GO TO 110
    WRITE(22,75)
    75    FORMAT(2X,'2 -1NM')
    WRITE(22,80)
    80    FORMAT(2X,'2 -1MC')
    WRITE(22,85)
    85    FORMAT(2X,'1 -1T')
    WRITE(22,90)
    90    FORMAT(2X,'2 -1MA')
```

```
GEN76170
GEN76180
GEN76190
GEN76200
GEN76210
GEN76220
GEN76230
GEN76240
GEN76250
GEN76260
GEN76270
GEN76280
GEN76290
GEN76300
GEN76310
GEN76320
GEN76330
GEN76340
GEN76350
GEN76360
GEN76370
GEN76380
GEN76390
GEN76400
GEN76410
GEN76420
GEN76430
GEN76440
GEN76450
GEN76460
GEN76470
GEN76480
GEN76490
GEN76500
GEN76510
GEN76520
GEN76530
GEN76540
GEN76550
GEN76560
GEN76570
GEN76580
GEN76590
GEN76600
GEN76610
GEN76620
GEN76630
GEN76640
GEN76650
GEN76660
GEN76670
GEN76680
GEN76690
GEN76700
GEN76710
GEN76720
GEN76730
GEN76740
GEN76750
GEN76760
GEN76770
GEN76780
GEN76790
GEN76800
```

```

      WRITE(22,95)
95      FORMAT(2X,'2 -1MO')

      WRITE(22,100)
100     FORMAT(2X,'2 -1PR')

      WRITE(22,105)
105     FORMAT(2X,'8 -1R MSNDEL')

110    RETURN
      END

C *****
C *
C *      THIS SUBROUTINE SOLVES FOR ALPHA GIVEN PHI
C *
C *****

      SUBROUTINE PALPSL(EE,RR,PI,PHI,TRANS,ALPHA,NO)

      FA(GAMMA)=TAN(PHI)*(EE*COS(3.0*GAMMA)+RR*COS(GAMMA))-
      #      (EE*SIN(3.0*GAMMA)+RR*SIN(GAMMA))+TRANS

      PHI=PHI*180.0/PI
      DO 5 II=1,11,2
        TEST=PHI*II
        IF(TEST.EQ.90.0*II) THEN

          PHI=PHI*PI/180.0
          ALPHA=PHI+0.10*PHI
          GO TO 25

        ENDIF

C *****
C *
C *      INCREMENTAL SEARCH METHOD
C *
C *****

5      CONTINUE

      EPSI=0.000001
      PHI=PHI*PI/180.0

      IT=0
      IL=0
      AAVG1=5.0

      DELTA=0.3*PI/180.0
      ALPHA=PHI

      AONE=FA(ALPHA)

10     ALPHA1=ALPHA+DELTA
      ATWO=FA(ALPHA1)

      IF(AONE*ATWO.EQ.0.0) THEN

        ALPHA=ATWO
        GO TO 25
      
```

GEN76810  
GEN76820  
GEN76830  
GEN76840  
GEN76850  
GEN76860  
GEN76870  
GEN76880  
GEN76890  
GEN76900  
GEN76910  
GEN76920  
GEN76930  
GEN76940  
GEN76950  
GEN76960  
GEN76970  
GEN76980  
GEN76990  
GEN77000  
GEN77010  
GEN77020  
GEN77030  
GEN77040  
GEN77050  
GEN77060  
GEN77070  
GEN77080  
GEN77090  
GEN77100  
GEN77110  
GEN77120  
GEN77130  
GEN77140  
GEN77150  
GEN77160  
GEN77170  
GEN77180  
GEN77190  
GEN77200  
GEN77210  
GEN77220  
GEN77230  
GEN77240  
GEN77250  
GEN77260  
GEN77270  
GEN77280  
GEN77290  
GEN77300  
GEN77310  
GEN77320  
GEN77330  
GEN77340  
GEN77350  
GEN77360  
GEN77370  
GEN77380  
GEN77390  
GEN77400  
GEN77410  
GEN77420  
GEN77430  
GEN77440

```
      ELSE IF (ATWO*AONE.GT.0.0) THEN
    IF (ATWO.GT.25.0) THEN
    WRITE(NO,*) 'ALPHA IS GOING TO INFINITY'
    STOP
    ELSE
      IT=IT+1
      IF ((ATWO.GT.0.0).AND.(AONE.GT.0.0)) THEN
        IF (AONE-ATWO.LT.0.0) THEN
          IL=IL+1
          IF (IL.GT.1) GO TO 15
          DELTA=-DELTA
        ENDIF
      ELSE IF ((ATWO.LT.0.0).AND.(AONE.LT.0.0)) THEN
        IF (AONE-ATWO.GT.0.0) THEN
          IL=IL+1
          IF (IL.GT.1) GO TO 15
          DELTA=-DELTA
        ENDIF
      ENDIF
    15  AONE=ATWO
        ALPHA=ALPHA1
        GO TO 10
    ENDIF
      ELSE IF (ATWO*AONE.LT.0.0) THEN
C *****
C *
C *          BISECTION METHOD
C *
C *****
    20  AAVG=(ALPHA+ALPHA1)/2.0
        ATHREE=FA(AAVG)
        IF (ATHREE.GT.10.0) THEN
          STOP
        ENDIF
        IF (ABS(AAVG1-AAVG).GT.EPSI) THEN
          IF (AONE*ATHREE.GT.0.0) THEN
            AAVG1=AAVG
            ALPHA=AAVG
            AONE=ATHREE
            GO TO 20
          ELSE IF (AONE*ATHREE.LT.0.0) THEN
```

```
GEN77450
GEN77460
GEN77470
GEN77480
GEN77490
GEN77500
GEN77510
GEN77520
GEN77530
GEN77540
GEN77550
GEN77560
GEN77570
GEN77580
GEN77590
GEN77600
GEN77610
GEN77620
GEN77630
GEN77640
GEN77650
GEN77660
GEN77670
GEN77680
GEN77690
GEN77700
GEN77710
GEN77720
GEN77730
GEN77740
GEN77750
GEN77760
GEN77770
GEN77780
GEN77790
GEN77800
GEN77810
GEN77820
GEN77830
GEN77840
GEN77850
GEN77860
GEN77870
GEN77880
GEN77890
GEN77900
GEN77910
GEN77920
GEN77930
GEN77940
GEN77950
GEN77960
GEN77970
GEN77980
GEN77990
GEN78000
GEN78010
GEN78020
GEN78030
GEN78040
GEN78050
GEN78060
GEN78070
GEN78080
```

```
        ATWO=ATHREE
        GO TO 20
    ENDIF
```

```
ELSE IF (ABS(AAVG1-AAVG).LT.EPSI) THEN
```

```
    ALPHA=AAVG
ENDIF
ENDIF
```

```
25  RETURN
    END
```

```
C *****
C *
C *   THIS SUBROUTINE SOLVES FOR ALPHA GIVEN A Y-COORDINATE *
C *
C *****
```

```
    SUBROUTINE YALPSL (EE,RR,Y,ICLK,ALPHA,NO)
```

```
    FA (GAMMA) =EE*SIN (3.0*GAMMA) +RR*SIN (GAMMA) -Y
```

```
    IT=0
    IL=0
    AAVG1=5.0
    EPSI=0.0001
```

```
    DELTA=1.0*3.141592654/180.0
```

```
    IF (ICLK.EQ.0) THEN
        ALPHA=0.1
```

```
    ELSE IF (ICLK.EQ.1) THEN
```

```
        ALPHA=2.75
    ENDIF
```

```
    AONE=FA (ALPHA)
```

```
C *****
C *
C *   INCREMENTAL SEARCH METHOD *
C *
C *****
```

```
5    ALPHA1=ALPHA+DELTA
    ATWO=FA (ALPHA1)
```

```
    IF (AONE*ATWO.EQ.0.0) THEN
```

```
        ALPHA=ATWO
        GO TO 20
```

```
    ELSE IF (ATWO*AONE.GT.0.0) THEN
```

```
    IF (ATWO.GT.25.0) THEN
        WRITE (NO,*) 'ALPHA IS GOING TO INFINITY'
        STOP
```

```
    ELSE
        IT=IT+1
```

GEN78090  
GEN78100  
GEN78110  
GEN78120  
GEN78130  
GEN78140  
GEN78150  
GEN78160  
GEN78170  
GEN78180  
GEN78190  
GEN78200  
GEN78210  
GEN78220  
GEN78230  
GEN78240  
GEN78250  
GEN78260  
GEN78270  
GEN78280  
GEN78290  
GEN78300  
GEN78310  
GEN78320  
GEN78330  
GEN78340  
GEN78350  
GEN78360  
GEN78370  
GEN78380  
GEN78390  
GEN78400  
GEN78410  
GEN78420  
GEN78430  
GEN78440  
GEN78450  
GEN78460  
GEN78470  
GEN78480  
GEN78490  
GEN78500  
GEN78510  
GEN78520  
GEN78530  
GEN78540  
GEN78550  
GEN78560  
GEN78570  
GEN78580  
GEN78590  
GEN78600  
GEN78610  
GEN78620  
GEN78630  
GEN78640  
GEN78650  
GEN78660  
GEN78670  
GEN78680  
GEN78690  
GEN78700  
GEN78710  
GEN78720

```
IF ((ATWO.GT.0.0).AND.(AONE.GT.0.0)) THEN
    IF (AONE-ATWO.LT.0.0) THEN
        IL=IL+1
        IF (IL.GT.1) GO TO 10
        DELTA=-DELTA
    ENDIF
ELSE IF ((ATWO.LT.0.0).AND.(AONE.LT.0.0)) THEN
    IF (AONE-ATWO.GT.0.0) THEN
        IL=IL+1
        IF (IL.GT.1) GO TO 10
        DELTA=-DELTA
    ENDIF
ENDIF
10  AONE=ATWO
    ALPHA=ALPHA1
    GO TO 5
ENDIF
ELSE IF (ATWO*AONE.LT.0.0) THEN
C *****
C *
C *          BISECTION METHOD
C *
C *****
15  AAVG=(ALPHA+ALPHA1)/2.0
    ATHREE=FA(AAVG)
    IF (ATHREE.GT.10.0) THEN
        STOP
    ENDIF
    IF (ABS(AAVG1-AAVG).GT.EPSI) THEN
        IF (AONE*ATHREE.GT.0.0) THEN
            AAVG1=AAVG
            ALPHA=AAVG
            AONE=ATHREE
            GO TO 15
        ELSE IF (AONE*ATHREE.LT.0.0) THEN
            AAVG1=AAVG
            ALPHA1=AAVG
            ATWO=ATHREE
            GO TO 15
        ENDIF
    ELSE IF (ABS(AAVG1-AAVG).LT.EPSI) THEN
        ALPHA=AAVG
    ENDIF
ENDIF
```

GEN78730  
GEN78740  
GEN78750  
GEN78760  
GEN78770  
GEN78780  
GEN78790  
GEN78800  
GEN78810  
GEN78820  
GEN78830  
GEN78840  
GEN78850  
GEN78860  
GEN78870  
GEN78880  
GEN78890  
GEN78900  
GEN78910  
GEN78920  
GEN78930  
GEN78940  
GEN78950  
GEN78960  
GEN78970  
GEN78980  
GEN78990  
GEN79000  
GEN79010  
GEN79020  
GEN79030  
GEN79040  
GEN79050  
GEN79060  
GEN79070  
GEN79080  
GEN79090  
GEN79100  
GEN79110  
GEN79120  
GEN79130  
GEN79140  
GEN79150  
GEN79160  
GEN79170  
GEN79180  
GEN79190  
GEN79200  
GEN79210  
GEN79220  
GEN79230  
GEN79240  
GEN79250  
GEN79260  
GEN79270  
GEN79280  
GEN79290  
GEN79300  
GEN79310  
GEN79320  
GEN79330  
GEN79340  
GEN79350  
GEN79360

```
20  RETURN
    END

C  *****
C  *
C  *      THIS SUBROUTINE SOLVES FOR THETA GIVEN ALPHA
C  *
C  *****

    SUBROUTINE THTASL (EE,RR,PI,ALPHA,THETA,NO)

        FA (GAMMA) = (RR+3.0*EE*COS (2.0*ALPHA)) / (SQRT
#      (9.0*EE**2+RR**2+6.0*EE*RR*COS (2.0*ALPHA))) -COS (GAMMA)

        IT=0
        IL=0
        TAVG1=5.0
        EPSI=0.0001

        DELTA=1.0*PI/180.0
        THETA=0.25

        TONE=FA (THETA)

C  *****
C  *
C  *      INCREMENTAL SEARCH METHOD
C  *
C  *****

5  THETA1=THETA+DELTA
    TTWO=FA (THETA1)

    IF (TONE*TTWO.EQ.0.0) THEN

        THETA=TTWO
        GO TO 20

    ELSE IF (TTWO*TONE.GT.0.0) THEN

        IF (TTWO.GT.25.0) THEN
            WRITE (NO,*) 'THETA IS GOING TO INFINITY'
            STOP
        ELSE
            IT=IT+1

            IF ((TTWO.GT.0.0).AND.(TONE.GT.0.0)) THEN

                IF (TONE-TTWO.LT.0.0) THEN
                    IL=IL+1
                    IF (IL.GT.1) GO TO 10

                    DELTA=-DELTA
                ENDIF
            ELSE IF ((TTWO.LT.0.0).AND.(TONE.LT.0.0)) THEN

                IF (TONE-TTWO.GT.0.0) THEN
                    IL=IL+1
                    IF (IL.GT.1) GO TO 10

                    DELTA=-DELTA
                ENDIF
            ENDIF
        ENDIF
    ENDIF

    GEN79370
    GEN79380
    GEN79390
    GEN79400
    GEN79410
    GEN79420
    GEN79430
    GEN79440
    GEN79450
    GEN79460
    GEN79470
    GEN79480
    GEN79490
    GEN79500
    GEN79510
    GEN79520
    GEN79530
    GEN79540
    GEN79550
    GEN79560
    GEN79570
    GEN79580
    GEN79590
    GEN79600
    GEN79610
    GEN79620
    GEN79630
    GEN79640
    GEN79650
    GEN79660
    GEN79670
    GEN79680
    GEN79690
    GEN79700
    GEN79710
    GEN79720
    GEN79730
    GEN79740
    GEN79750
    GEN79760
    GEN79770
    GEN79780
    GEN79790
    GEN79800
    GEN79810
    GEN79820
    GEN79830
    GEN79840
    GEN79850
    GEN79860
    GEN79870
    GEN79880
    GEN79890
    GEN79900
    GEN79910
    GEN79920
    GEN79930
    GEN79940
    GEN79950
    GEN79960
    GEN79970
    GEN79980
    GEN79990
    GEN80000
```

```

      ENDIF
10    TONE=TTWO
      THETA=THETA1
      GO TO 5
      ENDIF
      ELSE IF (TTWO*TONE.LT.0.0) THEN
C *****
C *
C *          BISECTION METHOD
C *
C *****
15    TAVG=(THETA+THETA1)/2.0
      TTHREE=FA(TAVG)
      IF (TTHREE.GT.10.0) THEN
          STOP
      ENDIF
      IF (ABS(TAVG1-TAVG).GT.EPSI) THEN
          IF (TONE*TTHREE.GT.0.0) THEN
              TAVG1=TAVG
              THETA=TAVG
              TONE=TTHREE
              GO TO 15
          ELSE IF (TONE*TTHREE.LT.0.0) THEN
              TAVG1=TAVG
              THETA1=TAVG
              TTWO=TTHREE
              GO TO 15
          ENDIF
      ELSE IF (ABS(TAVG1-TAVG).LT.EPSI) THEN
          THETA=TAVG
      ENDIF
      ENDIF
20    IF (((ALPHA.GT.0.0).AND.(ALPHA.LT.PI/2.0)).OR.((ALPHA.GT.PI).AND.
#      (ALPHA.LT.3.0*PI/2.0))) THETA=-THETA
      RETURN
      END
C *****
C *
C *          THIS SUBROUTINE CALCULATES THE POINT COORDINATES
C *          OF THE EXHAUST PORT
C *
C *****
      SUBROUTINE EXHST (EE,RR,R,PI,REXPT,TEXPT,AA,BB,CC,D,DEPTH,
#      TRANS,PHI1EP,PHI2EP,PHI1OD,PHI2OD,PHI1NW,
#      GIX,GIY,GGIX,GGIY,GGGIX,GGGIY)
      GEN80010
      GEN80020
      GEN80030
      GEN80040
      GEN80050
      GEN80060
      GEN80070
      GEN80080
      GEN80090
      GEN80100
      GEN80110
      GEN80120
      GEN80130
      GEN80140
      GEN80150
      GEN80160
      GEN80170
      GEN80180
      GEN80190
      GEN80200
      GEN80210
      GEN80220
      GEN80230
      GEN80240
      GEN80250
      GEN80260
      GEN80270
      GEN80280
      GEN80290
      GEN80300
      GEN80310
      GEN80320
      GEN80330
      GEN80340
      GEN80350
      GEN80360
      GEN80370
      GEN80380
      GEN80390
      GEN80400
      GEN80410
      GEN80420
      GEN80430
      GEN80440
      GEN80450
      GEN80460
      GEN80470
      GEN80480
      GEN80490
      GEN80500
      GEN80510
      GEN80520
      GEN80530
      GEN80540
      GEN80550
      GEN80560
      GEN80570
      GEN80580
      GEN80590
      GEN80600
      GEN80610
      GEN80620
      GEN80630
      GEN80640
```



```
COMMON / PORT / X1(7),Y1(7),Z1(7),X2(7),Y2(7),Z2(7),X3(7),Y3(7),
# Z3(7),X4(7),Y4(7),Z4(7),X5(7),Y5(7),Z5(7),X6(7),Y6(7),Z6(7),
# X7(7),Y7(7),Z7(7),X8(7),Y8(7),Z8(7),X9(7),Y9(7),Z9(7),
# X10(7),Y10(7),Z10(7),X11(7),Y11(7),Z11(7),X12(7),Y12(7),Z12(7),
# X13(7),Y13(7),Z13(7),X14(7),Y14(7),Z14(7),X15(7),Y15(7),Z15(7),
# X16(7),Y16(7),Z16(7),X17(7),Y17(7),Z17(7),X18(7),Y18(7),Z18(7),
# X19(7),Y19(7),Z19(7),X20(7),Y20(7),Z20(7),X21(7),Y21(7),Z21(7),
# X22(7),Y22(7),Z22(7),X23(7),Y23(7),Z23(7),X24(7),Y24(7),Z24(7),
# X25(7),Y25(7),Z25(7),X26(7),Y26(7),Z26(7),X27(7),Y27(7),Z27(7),
# X28(7),Y28(7),Z28(7),X29(7),Y29(7),Z29(7),X30(7),Y30(7),Z30(7),
# X31(7),Y31(7),Z31(7),X32(7),Y32(7),Z32(7),X33(7),Y33(7),Z33(7),
# X34(7),Y34(7),Z34(7),X35(7),Y35(7),Z35(7),X36(7),Y36(7),Z36(7),
# X37(7),Y37(7),Z37(7),X38(7),Y38(7),Z38(7),X39(7),Y39(7),Z39(7),
# X40(7),Y40(7),Z40(7),X41(7),Y41(7),Z41(7),X42(7),Y42(7),Z42(7),
# X43(7),Y43(7),Z43(7),X44(7),Y44(7),Z44(7),X45(7),Y45(7),Z45(7),
# X46(7),Y46(7),Z46(7),X47(7),Y47(7),Z47(7),X48(7),Y48(7),Z48(7),
# X49(7),Y49(7),Z49(7),X50(7),Y50(7),Z50(7),X51(7),Y51(7),Z51(7),
# X52(7),Y52(7),Z52(7),X53(7),Y53(7),Z53(7)

XXX(Z,BETA)=Z*COS(BETA)

YYY(ZZ,BETA)=ZZ*SIN(BETA)

FNX(GAMMA)=EE*COS(3.0*GAMMA)+RR*COS(GAMMA)

FNY(GAMMA)=EE*SIN(3.0*GAMMA)+RR*SIN(GAMMA)

FINERX(GAMMA,RLAMB,FF)=EE*COS(3.0*GAMMA)+RR*COS(GAMMA)+
# FF*COS(GAMMA+RLAMB)

FINERY(GAMMA,RLAMB,FF)=EE*SIN(3.0*GAMMA)+RR*SIN(GAMMA)+
# FF*SIN(GAMMA+RLAMB)

R=RR-EE+D-TRANS+AA
TRANS=-TRANS
PHI=(PHI1EP+PHI2EP)/2.0
DROP=R*SIN(PHI)
YCNTR=TRANS-DROP
YYONE=R*SIN(PHI1EP)
YYTWO=R*SIN(PHI2EP)
YYTRE=(YYONE-YYTWO)/2.0

DO 15 JJ=1,7

Y1(JJ)=YCNTR-REXPT
Z1(JJ)=DEPTH/2.0
X1(JJ)=SQRT(R**2-(Y1(JJ)-TRANS)**2)

Y2(JJ)=YCNTR-REXPT*SIN(60.0*PI/180.0)
Z2(JJ)=DEPTH/2.0+REXPT*COS(60.0*PI/180.0)
X2(JJ)=SQRT(R**2-(Y2(JJ)-TRANS)**2)

Y3(JJ)=YCNTR-REXPT*SIN(30.0*PI/180.0)
Z3(JJ)=DEPTH/2.0+REXPT*COS(30.0*PI/180.0)
X3(JJ)=SQRT(R**2-(Y3(JJ)-TRANS)**2)

Y4(JJ)=YCNTR
Z4(JJ)=DEPTH/2.0+REXPT
X4(JJ)=SQRT(R**2-(Y4(JJ)-TRANS)**2)

Y5(JJ)=YCNTR+REXPT*SIN(30.0*PI/180.0)
Z5(JJ)=DEPTH/2.0+REXPT*COS(30.0*PI/180.0)
```

```
X5(JJ)=SQRT(R**2-(Y5(JJ)-TRANS)**2)
GEN81290
GEN81300
Y6(JJ)=YCNTR+REXPT*SIN(60.0*PI/180.0)
GEN81310
Z6(JJ)=DEPTH/2.0+REXPT*COS(60.0*PI/180.0)
GEN81320
X6(JJ)=SQRT(R**2-(Y6(JJ)-TRANS)**2)
GEN81330
GEN81340
Y7(JJ)=YCNTR+REXPT
GEN81350
Z7(JJ)=DEPTH/2.0
GEN81360
X7(JJ)=SQRT(R**2-(Y7(JJ)-TRANS)**2)
GEN81370
GEN81380
Y8(JJ)=YCNTR+REXPT*SIN(60.0*PI/180.0)
GEN81390
Z8(JJ)=DEPTH/2.0-REXPT*COS(60.0*PI/180.0)
GEN81400
X8(JJ)=SQRT(R**2-(Y8(JJ)-TRANS)**2)
GEN81410
GEN81420
Y9(JJ)=YCNTR+REXPT*SIN(30.0*PI/180.0)
GEN81430
Z9(JJ)=DEPTH/2.0-REXPT*COS(30.0*PI/180.0)
GEN81440
X9(JJ)=SQRT(R**2-(Y9(JJ)-TRANS)**2)
GEN81450
GEN81460
Y10(JJ)=YCNTR
GEN81470
Z10(JJ)=DEPTH/2.0-REXPT
GEN81480
X10(JJ)=SQRT(R**2-(Y10(JJ)-TRANS)**2)
GEN81490
GEN81500
Y11(JJ)=YCNTR-REXPT*SIN(30.0*PI/180.0)
GEN81510
Z11(JJ)=DEPTH/2.0-REXPT*COS(30.0*PI/180.0)
GEN81520
X11(JJ)=SQRT(R**2-(Y11(JJ)-TRANS)**2)
GEN81530
GEN81540
Y12(JJ)=YCNTR-REXPT*SIN(60.0*PI/180.0)
GEN81550
Z12(JJ)=DEPTH/2.0-REXPT*COS(60.0*PI/180.0)
GEN81560
X12(JJ)=SQRT(R**2-(Y12(JJ)-TRANS)**2)
GEN81570
GEN81580
Y13(JJ)=YCNTR-REXPT-TEXPT
GEN81590
X13(JJ)=SQRT(R**2-(Y13(JJ)-TRANS)**2)
GEN81600
Z13(JJ)=3.0*DEPTH/4.0
GEN81610
GEN81620
IF(JJ.EQ.5) THEN
GEN81630
GEN81640
GEN81650
Y13(JJ)=GIY
GEN81660
X13(JJ)=GIX
GEN81670
ELSE IF (JJ.EQ.6) THEN
GEN81680
GEN81690
Y13(JJ)=GGIY
GEN81700
X13(JJ)=GGIX
GEN81710
ELSE IF (JJ.EQ.7) THEN
GEN81720
GEN81730
Y13(JJ)=GGGIY
GEN81740
X13(JJ)=GGGIX
GEN81750
GEN81760
ENDIF
GEN81770
GEN81780
X14(JJ)=X13(JJ)
GEN81790
Y14(JJ)=Y13(JJ)
GEN81800
Z14(JJ)=DEPTH/2.0
GEN81810
GEN81820
X15(JJ)=X13(JJ)
GEN81830
Y15(JJ)=Y13(JJ)
GEN81840
Z15(JJ)=DEPTH/4.0
GEN81850
GEN81860
Y16(JJ)=Y3(JJ)-TEXPT*SIN(30.0*PI/180.0)
GEN81870
Z16(JJ)=Z3(JJ)+TEXPT*COS(30.0*PI/180.0)
GEN81880
X16(JJ)=SQRT(R**2-(Y16(JJ)-TRANS)**2)
GEN81890
GEN81900
Y17(JJ)=YCNTR
GEN81910
Z17(JJ)=Z4(JJ)+TEXPT
GEN81920
```

```
X17(JJ)=SQRT(R**2-(Y17(JJ)-TRANS)**2)      GEN81930
                                           GEN81940
Y18(JJ)=Y5(JJ)+TEXPT*SIN(30.0*PI/180.0)    GEN81950
Z18(JJ)=Z5(JJ)+TEXPT*COS(30.0*PI/180.0)    GEN81960
X18(JJ)=SQRT(R**2-(Y18(JJ)-TRANS)**2)      GEN81970
                                           GEN81980
X19(JJ)=XXX(R,PHI1NW)                      GEN81990
Y19(JJ)=-YYY(R,PHI1NW)+TRANS               GEN82000
Z19(JJ)=DEPTH/4.0                         GEN82010
                                           GEN82020
X20(JJ)=X19(JJ)                           GEN82030
Y20(JJ)=Y19(JJ)                           GEN82040
Z20(JJ)=DEPTH/2.0                         GEN82050
                                           GEN82060
X21(JJ)=X19(JJ)                           GEN82070
Y21(JJ)=Y19(JJ)                           GEN82080
Z21(JJ)=3.0*DEPTH/4.0                    GEN82090
                                           GEN82100
Y22(JJ)=Y9(JJ)+TEXPT*SIN(30.0*PI/180.0)    GEN82110
Z22(JJ)=Z9(JJ)-TEXPT*COS(30.0*PI/180.0)    GEN82120
X22(JJ)=SQRT(R**2-(Y22(JJ)-TRANS)**2)      GEN82130
                                           GEN82140
Y23(JJ)=YCNR                              GEN82150
Z23(JJ)=Z10(JJ)-TEXPT                    GEN82160
X23(JJ)=SQRT(R**2-(Y23(JJ)-TRANS)**2)      GEN82170
                                           GEN82180
Y24(JJ)=Y11(JJ)-TEXPT*SIN(30.0*PI/180.0)    GEN82190
Z24(JJ)=Z11(JJ)-TEXPT*COS(30.0*PI/180.0)    GEN82200
X24(JJ)=SQRT(R**2-(Y24(JJ)-TRANS)**2)      GEN82210
                                           GEN82220
Z25(JJ)=0.0                             GEN82230
Z26(JJ)=0.0                             GEN82240
Z27(JJ)=0.0                             GEN82250
Z28(JJ)=0.0                             GEN82260
Z31(JJ)=DEPTH                            GEN82270
Z32(JJ)=DEPTH                            GEN82280
Z33(JJ)=DEPTH                            GEN82290
Z34(JJ)=DEPTH                            GEN82300
                                           GEN82310
      IF ((JJ.EQ.1).OR.(JJ.EQ.2).OR.(JJ.EQ.5).OR.(JJ.EQ.6).
#      OR.(JJ.EQ.7)) THEN                  GEN82320
                                           GEN82330
Y26(JJ)=Y16(JJ)                          GEN82340
X26(JJ)=X16(JJ)                          GEN82350
                                           GEN82360
Y27(JJ)=Y17(JJ)                          GEN82370
X27(JJ)=X17(JJ)                          GEN82380
                                           GEN82390
Y28(JJ)=Y18(JJ)                          GEN82400
X28(JJ)=X18(JJ)                          GEN82410
                                           GEN82420
Y31(JJ)=Y22(JJ)                          GEN82430
X31(JJ)=X22(JJ)                          GEN82440
                                           GEN82450
Y32(JJ)=Y23(JJ)                          GEN82460
X32(JJ)=X23(JJ)                          GEN82470
                                           GEN82480
X33(JJ)=X24(JJ)                          GEN82490
Y33(JJ)=Y24(JJ)                          GEN82500
                                           GEN82510
X34(JJ)=X13(JJ)                          GEN82520
Y34(JJ)=Y13(JJ)                          GEN82530
                                           GEN82540
X25(JJ)=X13(JJ)                          GEN82550
                                           GEN82560
```

```
Y25(JJ)=Y13(JJ)
ENDIF
X29(JJ)=X19(JJ)
Y29(JJ)=Y19(JJ)
Z29(JJ)=0.0
X30(JJ)=X29(JJ)
Y30(JJ)=Y29(JJ)
Z30(JJ)=DEPTH
Z35(JJ)=DEPTH
Z36(JJ)=3.0*DEPTH/4.0
Z37(JJ)=DEPTH/2.0
Z38(JJ)=DEPTH/4.0
Z39(JJ)=0.0
Z40(JJ)=DEPTH
Z41(JJ)=3.0*DEPTH/4.0
Z42(JJ)=DEPTH/2.0
Z43(JJ)=DEPTH/4.0
Z44(JJ)=0.0
Z45(JJ)=DEPTH
Z46(JJ)=3.0*DEPTH/4.0
Z47(JJ)=DEPTH/2.0
Z48(JJ)=DEPTH/4.0
Z49(JJ)=0.0
IF((JJ.EQ.1).OR.(JJ.EQ.2)) THEN
  PHI11=ATAN(ABS(Y34(1)-TRANS)/X34(1))
  PHI12=PHI2OD-PHI11
  PHI13=PHI11+2.0*PHI12/5.0
  PHI14=PHI11+3.0*PHI12/4.0
  X35(JJ)=XXX(R,PHI13)
  Y35(JJ)=-YYY(R,PHI13)+TRANS
  X40(JJ)=XXX(R,PHI14)
  Y40(JJ)=-YYY(R,PHI14)+TRANS
  X45(JJ)=XXX(R,PHI2OD)
  Y45(JJ)=-YYY(R,PHI2OD)+TRANS
  X36(JJ)=X35(JJ)
  Y36(JJ)=Y35(JJ)
  X37(JJ)=X35(JJ)
  Y37(JJ)=Y35(JJ)
  X38(JJ)=X35(JJ)
  Y38(JJ)=Y35(JJ)
  X39(JJ)=X35(JJ)
  Y39(JJ)=Y35(JJ)
  X41(JJ)=X40(JJ)
  Y41(JJ)=Y40(JJ)
  X42(JJ)=X40(JJ)
  Y42(JJ)=Y40(JJ)
  X43(JJ)=X40(JJ)
  Y43(JJ)=Y40(JJ)
  X44(JJ)=X40(JJ)
  Y44(JJ)=Y40(JJ)
  X46(JJ)=X45(JJ)
  Y46(JJ)=Y45(JJ)
```

GEN82570  
GEN82580  
GEN82590  
GEN82600  
GEN82610  
GEN82620  
GEN82630  
GEN82640  
GEN82650  
GEN82660  
GEN82670  
GEN82680  
GEN82690  
GEN82700  
GEN82710  
GEN82720  
GEN82730  
GEN82740  
GEN82750  
GEN82760  
GEN82770  
GEN82780  
GEN82790  
GEN82800  
GEN82810  
GEN82820  
GEN82830  
GEN82840  
GEN82850  
GEN82860  
GEN82870  
GEN82880  
GEN82890  
GEN82900  
GEN82910  
GEN82920  
GEN82930  
GEN82940  
GEN82950  
GEN82960  
GEN82970  
GEN82980  
GEN82990  
GEN83000  
GEN83010  
GEN83020  
GEN83030  
GEN83040  
GEN83050  
GEN83060  
GEN83070  
GEN83080  
GEN83090  
GEN83100  
GEN83110  
GEN83120  
GEN83130  
GEN83140  
GEN83150  
GEN83160  
GEN83170  
GEN83180  
GEN83190  
GEN83200

```
X47(JJ)=X45(JJ)
Y47(JJ)=Y45(JJ)
X48(JJ)=X45(JJ)
Y48(JJ)=Y45(JJ)
X49(JJ)=X45(JJ)
Y49(JJ)=Y45(JJ)
```

```
ENDIF
```

```
IF(JJ.EQ.1) R=R-AA
IF(JJ.EQ.2) R=R-D/3.0
IF(JJ.EQ.3) R=R-D/3.0
IF(JJ.EQ.4) R=R-D/3.0
IF(JJ.EQ.5) R=R+BB
IF(JJ.EQ.6) R=R-BB+CC
```

```
15 CONTINUE
```

```
RETURN
END
```

```
C *****
C *
C * THIS SUBROUTINE CALCULATES THE POINT COORDINATES
C * OF THE INTAKE PORT
C *
C *****
```

```
SUBROUTINE INTKE (EE,RR,R,PI,RINPT,TINPT,AA,BB,CC,D,DEPTH,
# TRANS,PHI1P,PHI2IP,PHI2OD,PHI1NW,LINPT,WINPT)
```

```
COMMON / PORT / X1(7),Y1(7),Z1(7),X2(7),Y2(7),Z2(7),X3(7),Y3(7),
# Z3(7),X4(7),Y4(7),Z4(7),X5(7),Y5(7),Z5(7),X6(7),Y6(7),Z6(7),
# X7(7),Y7(7),Z7(7),X8(7),Y8(7),Z8(7),X9(7),Y9(7),Z9(7),
# X10(7),Y10(7),Z10(7),X11(7),Y11(7),Z11(7),X12(7),Y12(7),Z12(7),
# X13(7),Y13(7),Z13(7),X14(7),Y14(7),Z14(7),X15(7),Y15(7),Z15(7),
# X16(7),Y16(7),Z16(7),X17(7),Y17(7),Z17(7),X18(7),Y18(7),Z18(7),
# X19(7),Y19(7),Z19(7),X20(7),Y20(7),Z20(7),X21(7),Y21(7),Z21(7),
# X22(7),Y22(7),Z22(7),X23(7),Y23(7),Z23(7),X24(7),Y24(7),Z24(7),
# X25(7),Y25(7),Z25(7),X26(7),Y26(7),Z26(7),X27(7),Y27(7),Z27(7),
# X28(7),Y28(7),Z28(7),X29(7),Y29(7),Z29(7),X30(7),Y30(7),Z30(7),
# X31(7),Y31(7),Z31(7),X32(7),Y32(7),Z32(7),X33(7),Y33(7),Z33(7),
# X34(7),Y34(7),Z34(7),X35(7),Y35(7),Z35(7),X36(7),Y36(7),Z36(7),
# X37(7),Y37(7),Z37(7),X38(7),Y38(7),Z38(7),X39(7),Y39(7),Z39(7),
# X40(7),Y40(7),Z40(7),X41(7),Y41(7),Z41(7),X42(7),Y42(7),Z42(7),
# X43(7),Y43(7),Z43(7),X44(7),Y44(7),Z44(7),X45(7),Y45(7),Z45(7),
# X46(7),Y46(7),Z46(7),X47(7),Y47(7),Z47(7),X48(7),Y48(7),Z48(7),
# X49(7),Y49(7),Z49(7),X50(7),Y50(7),Z50(7),X51(7),Y51(7),Z51(7),
# X52(7),Y52(7),Z52(7),X53(7),Y53(7),Z53(7)
```

```
REAL LINPT
```

```
XXX(Z,BETA)=Z*COS(BETA)
```

```
YYY(ZZ,BETA)=ZZ*SIN(BETA)
```

```
TRANS=ABS(TRANS)
R=RR-EE+D+AA-TRANS
PHI=(PHI1P+PHI2IP)/2.0
PHIZ=PHI*180.0/PI
DROP=R*SIN(PHI)
```

```
GEN83210
GEN83220
GEN83230
GEN83240
GEN83250
GEN83260
GEN83270
GEN83280
GEN83290
GEN83300
GEN83310
GEN83320
GEN83330
GEN83340
GEN83350
GEN83360
GEN83370
GEN83380
GEN83390
GEN83400
GEN83410
GEN83420
GEN83430
GEN83440
GEN83450
GEN83460
GEN83470
GEN83480
GEN83490
GEN83500
GEN83510
GEN83520
GEN83530
GEN83540
GEN83550
GEN83560
GEN83570
GEN83580
GEN83590
GEN83600
GEN83610
GEN83620
GEN83630
GEN83640
GEN83650
GEN83660
GEN83670
GEN83680
GEN83690
GEN83700
GEN83710
GEN83720
GEN83730
GEN83740
GEN83750
GEN83760
GEN83770
GEN83780
GEN83790
GEN83800
GEN83810
GEN83820
GEN83830
GEN83840
```

```
YCNR=TRANS+DROP
YYONE=R*SIN(PHI1IP)
YYTWO=R*SIN(PHI2IP)
YYTRE=YYTWO-YYONE

DO 15 JJ=1,7

  IF ((JJ.EQ.1).OR.(JJ.EQ.2)) THEN

    Y1(JJ)=YCNR-RINPT
    Z1(JJ)=DEPTH/2.0
    X1(JJ)=SQRT(R**2-(Y1(JJ)-TRANS)**2)

    Y2(JJ)=YCNR-RINPT*SIN(60.0*PI/180.0)
    Z2(JJ)=DEPTH/2.0+RINPT*COS(60.0*PI/180.0)
    X2(JJ)=SQRT(R**2-(Y2(JJ)-TRANS)**2)

    Y3(JJ)=YCNR-RINPT*SIN(30.0*PI/180.0)
    Z3(JJ)=DEPTH/2.0+RINPT*COS(30.0*PI/180.0)
    X3(JJ)=SQRT(R**2-(Y3(JJ)-TRANS)**2)

    Y4(JJ)=YCNR
    Z4(JJ)=DEPTH/2.0+RINPT
    X4(JJ)=SQRT(R**2-(Y4(JJ)-TRANS)**2)

    Y5(JJ)=YCNR+RINPT*SIN(30.0*PI/180.0)
    Z5(JJ)=DEPTH/2.0+RINPT*COS(30.0*PI/180.0)
    X5(JJ)=SQRT(R**2-(Y5(JJ)-TRANS)**2)

    Y6(JJ)=YCNR+RINPT*SIN(60.0*PI/180.0)
    Z6(JJ)=DEPTH/2.0+RINPT*COS(60.0*PI/180.0)
    X6(JJ)=SQRT(R**2-(Y6(JJ)-TRANS)**2)

    Y7(JJ)=YCNR+RINPT
    Z7(JJ)=DEPTH/2.0
    X7(JJ)=SQRT(R**2-(Y7(JJ)-TRANS)**2)

    Y8(JJ)=YCNR+RINPT*SIN(60.0*PI/180.0)
    Z8(JJ)=DEPTH/2.0-RINPT*COS(60.0*PI/180.0)
    X8(JJ)=SQRT(R**2-(Y8(JJ)-TRANS)**2)

    Y9(JJ)=YCNR+RINPT*SIN(30.0*PI/180.0)
    Z9(JJ)=DEPTH/2.0-RINPT*COS(30.0*PI/180.0)
    X9(JJ)=SQRT(R**2-(Y9(JJ)-TRANS)**2)

    Y10(JJ)=YCNR
    Z10(JJ)=DEPTH/2.0-RINPT
    X10(JJ)=SQRT(R**2-(Y10(JJ)-TRANS)**2)

    Y11(JJ)=YCNR-RINPT*SIN(30.0*PI/180.0)
    Z11(JJ)=DEPTH/2.0-RINPT*COS(30.0*PI/180.0)
    X11(JJ)=SQRT(R**2-(Y11(JJ)-TRANS)**2)

    Y12(JJ)=YCNR-RINPT*SIN(60.0*PI/180.0)
    Z12(JJ)=DEPTH/2.0-RINPT*COS(60.0*PI/180.0)
    X12(JJ)=SQRT(R**2-(Y12(JJ)-TRANS)**2)

  ELSE IF (JJ.GT.2) THEN

    Y1(JJ)=YCNR-LINPT/2.0
    Z1(JJ)=DEPTH/2.0
    X1(JJ)=SQRT(R**2-(Y1(JJ)-TRANS)**2)

    Y2(JJ)=YCNR-LINPT/2.0
```

GEN83850  
GEN83860  
GEN83870  
GEN83880  
GEN83890  
GEN83900  
GEN83910  
GEN83920  
GEN83930  
GEN83940  
GEN83950  
GEN83960  
GEN83970  
GEN83980  
GEN83990  
GEN84000  
GEN84010  
GEN84020  
GEN84030  
GEN84040  
GEN84050  
GEN84060  
GEN84070  
GEN84080  
GEN84090  
GEN84100  
GEN84110  
GEN84120  
GEN84130  
GEN84140  
GEN84150  
GEN84160  
GEN84170  
GEN84180  
GEN84190  
GEN84200  
GEN84210  
GEN84220  
GEN84230  
GEN84240  
GEN84250  
GEN84260  
GEN84270  
GEN84280  
GEN84290  
GEN84300  
GEN84310  
GEN84320  
GEN84330  
GEN84340  
GEN84350  
GEN84360  
GEN84370  
GEN84380  
GEN84390  
GEN84400  
GEN84410  
GEN84420  
GEN84430  
GEN84440  
GEN84450  
GEN84460  
GEN84470  
GEN84480

```
      Z2(JJ)=DEPTH/2.0+WINPT/2.0      GEN84490
      X2(JJ)=SQRT(R**2-(Y2(JJ)-TRANS)**2) GEN84500
                                          GEN84510
      Y3(JJ)=YCNR-LINPT/4.0           GEN84520
      Z3(JJ)=DEPTH/2.0+WINPT/2.0      GEN84530
      X3(JJ)=SQRT(R**2-(Y3(JJ)-TRANS)**2) GEN84540
                                          GEN84550
      Y4(JJ)=YCNR                     GEN84560
      Z4(JJ)=DEPTH/2.0+WINPT/2.0      GEN84570
      X4(JJ)=SQRT(R**2-(Y4(JJ)-TRANS)**2) GEN84580
                                          GEN84590
      Y5(JJ)=YCNR+LINPT/4.0           GEN84600
      Z5(JJ)=DEPTH/2.0+WINPT/2.0      GEN84610
      X5(JJ)=SQRT(R**2-(Y5(JJ)-TRANS)**2) GEN84620
                                          GEN84630
      Y6(JJ)=YCNR+LINPT/2.0           GEN84640
      Z6(JJ)=DEPTH/2.0+WINPT/2.0      GEN84650
      X6(JJ)=SQRT(R**2-(Y6(JJ)-TRANS)**2) GEN84660
                                          GEN84670
      Y7(JJ)=YCNR+LINPT/2.0           GEN84680
      Z7(JJ)=DEPTH/2.0                GEN84690
      X7(JJ)=SQRT(R**2-(Y7(JJ)-TRANS)**2) GEN84700
                                          GEN84710
      Y8(JJ)=YCNR+LINPT/2.0           GEN84720
      Z8(JJ)=DEPTH/2.0-WINPT/2.0      GEN84730
      X8(JJ)=SQRT(R**2-(Y8(JJ)-TRANS)**2) GEN84740
                                          GEN84750
      Y9(JJ)=YCNR+LINPT/4.0           GEN84760
      Z9(JJ)=DEPTH/2.0-WINPT/2.0      GEN84770
      X9(JJ)=SQRT(R**2-(Y9(JJ)-TRANS)**2) GEN84780
                                          GEN84790
      Y10(JJ)=YCNR                    GEN84800
      Z10(JJ)=DEPTH/2.0-WINPT/2.0     GEN84810
      X10(JJ)=SQRT(R**2-(Y10(JJ)-TRANS)**2) GEN84820
                                          GEN84830
      Y11(JJ)=YCNR-LINPT/4.0          GEN84840
      Z11(JJ)=DEPTH/2.0-WINPT/2.0     GEN84850
      X11(JJ)=SQRT(R**2-(Y11(JJ)-TRANS)**2) GEN84860
                                          GEN84870
      Y12(JJ)=YCNR-LINPT/2.0          GEN84880
      Z12(JJ)=DEPTH/2.0-WINPT/2.0     GEN84890
      X12(JJ)=SQRT(R**2-(Y12(JJ)-TRANS)**2) GEN84900
                                          GEN84910
      ENDIF                            GEN84920
                                          GEN84930
      X13(JJ)=XXX(R,PHI1P)             GEN84940
      Y13(JJ)=YYY(R,PHI1P)+TRANS      GEN84950
      Z13(JJ)=3.0*DEPTH/4.0           GEN84960
                                          GEN84970
      X14(JJ)=X13(JJ)                 GEN84980
      Y14(JJ)=Y13(JJ)                 GEN84990
      Z14(JJ)=DEPTH/2.0               GEN85000
                                          GEN85010
      X15(JJ)=X13(JJ)                 GEN85020
      Y15(JJ)=Y13(JJ)                 GEN85030
      Z15(JJ)=DEPTH/4.0               GEN85040
                                          GEN85050
      IF ((JJ.EQ.1).OR.(JJ.EQ.2)) THEN GEN85060
                                          GEN85070
      Y17(JJ)=Y3(JJ)-TINPT*SIN(30.0*PI/180.0) GEN85080
      Z17(JJ)=Z3(JJ)+TINPT*COS(30.0*PI/180.0) GEN85090
      X17(JJ)=SQRT(R**2-(Y17(JJ)-TRANS)**2) GEN85100
                                          GEN85110
      Y18(JJ)=YCNR                    GEN85120
```

```
      Z18(JJ)=Z4(JJ)+TINPT
      X18(JJ)=SQRT(R**2-(Y18(JJ)-TRANS)**2)

      Y19(JJ)=Y5(JJ)+TINPT*SIN(30.0*PI/180.0)
      Z19(JJ)=Z5(JJ)+TINPT*COS(30.0*PI/180.0)
      X19(JJ)=SQRT(R**2-(Y19(JJ)-TRANS)**2)

    ELSE IF (JJ.GT.2) THEN

      Y17(JJ)=Y3(JJ)
      Z17(JJ)=Z3(JJ)+TINPT
      X17(JJ)=X3(JJ)

      Y18(JJ)=YCNTR
      Z18(JJ)=Z4(JJ)+TINPT
      X18(JJ)=X4(JJ)

      Y19(JJ)=Y5(JJ)
      Z19(JJ)=Z5(JJ)+TINPT
      X19(JJ)=X5(JJ)

    ENDIF

      X16(JJ)=X2(JJ)
      Y16(JJ)=Y2(JJ)
      Z16(JJ)=Z17(JJ)

      Y20(JJ)=YCNTR+RINPT+TINPT
      X20(JJ)=SQRT(R**2-(Y20(JJ)-TRANS)**2)
      Z20(JJ)=Z19(JJ)

      X21(JJ)=X20(JJ)
      Y21(JJ)=Y20(JJ)
      Z21(JJ)=DEPTH/2.0

    IF ((JJ.EQ.1).OR.(JJ.EQ.2)) THEN

      Y23(JJ)=Y9(JJ)+TINPT*SIN(30.0*PI/180.0)
      Z23(JJ)=Z9(JJ)-TINPT*COS(30.0*PI/180.0)
      X23(JJ)=SQRT(R**2-(Y23(JJ)-TRANS)**2)

      Y24(JJ)=YCNTR
      Z24(JJ)=Z10(JJ)-TINPT
      X24(JJ)=SQRT(R**2-(Y24(JJ)-TRANS)**2)

      Y25(JJ)=Y11(JJ)-TINPT*SIN(30.0*PI/180.0)
      Z25(JJ)=Z11(JJ)-TINPT*COS(30.0*PI/180.0)
      X25(JJ)=SQRT(R**2-(Y25(JJ)-TRANS)**2)

    ELSE IF (JJ.GT.2) THEN

      Y23(JJ)=Y9(JJ)
      Z23(JJ)=Z9(JJ)-TINPT
      X23(JJ)=X9(JJ)

      Y24(JJ)=YCNTR
      Z24(JJ)=Z10(JJ)-TINPT
      X24(JJ)=X10(JJ)

      Y25(JJ)=Y11(JJ)
      Z25(JJ)=Z11(JJ)-TINPT
      X25(JJ)=X11(JJ)

    ENDIF
```

GEN85130  
GEN85140  
GEN85150  
GEN85160  
GEN85170  
GEN85180  
GEN85190  
GEN85200  
GEN85210  
GEN85220  
GEN85230  
GEN85240  
GEN85250  
GEN85260  
GEN85270  
GEN85280  
GEN85290  
GEN85300  
GEN85310  
GEN85320  
GEN85330  
GEN85340  
GEN85350  
GEN85360  
GEN85370  
GEN85380  
GEN85390  
GEN85400  
GEN85410  
GEN85420  
GEN85430  
GEN85440  
GEN85450  
GEN85460  
GEN85470  
GEN85480  
GEN85490  
GEN85500  
GEN85510  
GEN85520  
GEN85530  
GEN85540  
GEN85550  
GEN85560  
GEN85570  
GEN85580  
GEN85590  
GEN85600  
GEN85610  
GEN85620  
GEN85630  
GEN85640  
GEN85650  
GEN85660  
GEN85670  
GEN85680  
GEN85690  
GEN85700  
GEN85710  
GEN85720  
GEN85730  
GEN85740  
GEN85750  
GEN85760



```
X22(JJ)=X20(JJ)
Y22(JJ)=Y20(JJ)
Z22(JJ)=Z23(JJ)

X26(JJ)=X12(JJ)
Y26(JJ)=Y12(JJ)
Z26(JJ)=Z25(JJ)

      IF ((JJ.EQ.1).OR.(JJ.EQ.2).OR.(JJ.EQ.5).OR.(JJ.EQ.6).
#      OR.(JJ.EQ.7)) THEN

      X27(JJ)=X13(JJ)
      Y27(JJ)=Y13(JJ)
      Z27(JJ)=0.0

      X28(JJ)=X16(JJ)
      Y28(JJ)=Y16(JJ)
      Z28(JJ)=0.0

      Y29(JJ)=Y17(JJ)
      X29(JJ)=X17(JJ)
      Z29(JJ)=0.0

      Y30(JJ)=Y18(JJ)
      X30(JJ)=X18(JJ)
      Z30(JJ)=0.0

      Y31(JJ)=Y19(JJ)
      X31(JJ)=X19(JJ)
      Z31(JJ)=0.0

      X32(JJ)=X20(JJ)
      Y32(JJ)=Y20(JJ)
      Z32(JJ)=0.0

      X33(JJ)=X22(JJ)
      Y33(JJ)=Y22(JJ)
      Z33(JJ)=DEPTH

      Y34(JJ)=Y23(JJ)
      X34(JJ)=X23(JJ)
      Z34(JJ)=DEPTH

      Y35(JJ)=Y24(JJ)
      X35(JJ)=X24(JJ)
      Z35(JJ)=DEPTH

      X36(JJ)=X25(JJ)
      Y36(JJ)=Y25(JJ)
      Z36(JJ)=DEPTH

      X37(JJ)=X26(JJ)
      Y37(JJ)=Y26(JJ)
      Z37(JJ)=DEPTH

      X38(JJ)=X13(JJ)
      Y38(JJ)=Y13(JJ)
      Z38(JJ)=DEPTH

      ENDIF

      IF ((JJ.NE.3).OR.(JJ.NE.4)) THEN
```

```
GEN85770
GEN85780
GEN85790
GEN85800
GEN85810
GEN85820
GEN85830
GEN85840
GEN85850
GEN85860
GEN85870
GEN85880
GEN85890
GEN85900
GEN85910
GEN85920
GEN85930
GEN85940
GEN85950
GEN85960
GEN85970
GEN85980
GEN85990
GEN86000
GEN86010
GEN86020
GEN86030
GEN86040
GEN86050
GEN86060
GEN86070
GEN86080
GEN86090
GEN86100
GEN86110
GEN86120
GEN86130
GEN86140
GEN86150
GEN86160
GEN86170
GEN86180
GEN86190
GEN86200
GEN86210
GEN86220
GEN86230
GEN86240
GEN86250
GEN86260
GEN86270
GEN86280
GEN86290
GEN86300
GEN86310
GEN86320
GEN86330
GEN86340
GEN86350
GEN86360
GEN86370
GEN86380
GEN86390
GEN86400
```

```
      X49(JJ)=XXX(R,PHI1NW)
      Y49(JJ)=YYY(R,PHI1NW)+TRANS
      Z49(JJ)=DEPTH

      IF (JJ.EQ.1) THEN

        PHI12=PHI1NW-PHI2IP
        PHI13=PHI2IP+1.0*PHI12/3.0
        PHI14=PHI2IP+2.0*PHI12/3.0

      ELSE IF (JJ.GT.1) THEN

        PHI12=PHI1NW-ATAN((Y21(JJ)-TRANS)/X21(JJ))
        PHI13=PHI1NW-2.0*PHI12/3.0
        PHI14=PHI1NW-1.0*PHI12/3.0

      ENDIF

      X39(JJ)=XXX(R,PHI13)
      Y39(JJ)=YYY(R,PHI13)+TRANS
      Z39(JJ)=DEPTH

      X40(JJ)=X39(JJ)
      Y40(JJ)=Y39(JJ)
      Z40(JJ)=3.0*DEPTH/4.0

      X41(JJ)=X39(JJ)
      Y41(JJ)=Y39(JJ)
      Z41(JJ)=DEPTH/2.0

      X42(JJ)=X39(JJ)
      Y42(JJ)=Y39(JJ)
      Z42(JJ)=DEPTH/4.0

      X43(JJ)=X39(JJ)
      Y43(JJ)=Y39(JJ)
      Z43(JJ)=0.0

      X44(JJ)=XXX(R,PHI14)
      Y44(JJ)=YYY(R,PHI14)+TRANS
      Z44(JJ)=DEPTH

      X45(JJ)=X44(JJ)
      Y45(JJ)=Y44(JJ)
      Z45(JJ)=3.0*DEPTH/4.0

      X46(JJ)=X44(JJ)
      Y46(JJ)=Y44(JJ)
      Z46(JJ)=DEPTH/2.0

      X47(JJ)=X44(JJ)
      Y47(JJ)=Y44(JJ)
      Z47(JJ)=DEPTH/4.0

      X48(JJ)=X44(JJ)
      Y48(JJ)=Y44(JJ)
      Z48(JJ)=0.0

      X50(JJ)=X49(JJ)
      Y50(JJ)=Y49(JJ)
      Z50(JJ)=3.0*DEPTH/4.0

      X51(JJ)=X49(JJ)
      Y51(JJ)=Y49(JJ)
```

```
GEN86410
GEN86420
GEN86430
GEN86440
GEN86450
GEN86460
GEN86470
GEN86480
GEN86490
GEN86500
GEN86510
GEN86520
GEN86530
GEN86540
GEN86550
GEN86560
GEN86570
GEN86580
GEN86590
GEN86600
GEN86610
GEN86620
GEN86630
GEN86640
GEN86650
GEN86660
GEN86670
GEN86680
GEN86690
GEN86700
GEN86710
GEN86720
GEN86730
GEN86740
GEN86750
GEN86760
GEN86770
GEN86780
GEN86790
GEN86800
GEN86810
GEN86820
GEN86830
GEN86840
GEN86850
GEN86860
GEN86870
GEN86880
GEN86890
GEN86900
GEN86910
GEN86920
GEN86930
GEN86940
GEN86950
GEN86960
GEN86970
GEN86980
GEN86990
GEN87000
GEN87010
GEN87020
GEN87030
GEN87040
```

```

      Z51(JJ)=DEPTH/2.0
      X52(JJ)=X49(JJ)
      Y52(JJ)=Y49(JJ)
      Z52(JJ)=DEPTH/4.0
      X53(JJ)=X49(JJ)
      Y53(JJ)=Y49(JJ)
      Z53(JJ)=0.0
      ENDIF
      IF(JJ.EQ.1) R=R-AA
      IF(JJ.EQ.2) R=R-D/3.0
      IF(JJ.EQ.3) R=R-D/3.0
      IF(JJ.EQ.4) R=R-D/3.0
      IF(JJ.EQ.5) R=R+BB
      IF(JJ.EQ.6) R=R-BB+CC
15  CONTINUE
      RETURN
      END
C *****
C *
C *   THIS SUBROUTINE CALCULATES THE POINT COORDINATES
C *   OF THE SPARK PLUG PORT
C *
C *****
      SUBROUTINE SPRKPG (EE,RR,R,PI,YONE,YTWO,PHONE,PHITWO,RSP,IEND,
#   AA,BBB,CC,D,DEPTH,TRANS,REGION,ICLK)
      COMMON / PORT / X1(7),Y1(7),Z1(7),X2(7),Y2(7),Z2(7),X3(7),Y3(7),
#   Z3(7),X4(7),Y4(7),Z4(7),X5(7),Y5(7),Z5(7),X6(7),Y6(7),Z6(7),
#   X7(7),Y7(7),Z7(7),X8(7),Y8(7),Z8(7),X9(7),Y9(7),Z9(7),
#   X10(7),Y10(7),Z10(7),X11(7),Y11(7),Z11(7),X12(7),Y12(7),Z12(7),
#   X13(7),Y13(7),Z13(7),X14(7),Y14(7),Z14(7),X15(7),Y15(7),Z15(7),
#   X16(7),Y16(7),Z16(7),X17(7),Y17(7),Z17(7),X18(7),Y18(7),Z18(7),
#   X19(7),Y19(7),Z19(7),X20(7),Y20(7),Z20(7),X21(7),Y21(7),Z21(7),
#   X22(7),Y22(7),Z22(7),X23(7),Y23(7),Z23(7),X24(7),Y24(7),Z24(7),
#   X25(7),Y25(7),Z25(7),X26(7),Y26(7),Z26(7),X27(7),Y27(7),Z27(7),
#   X28(7),Y28(7),Z28(7),X29(7),Y29(7),Z29(7),X30(7),Y30(7),Z30(7),
#   X31(7),Y31(7),Z31(7),X32(7),Y32(7),Z32(7),X33(7),Y33(7),Z33(7),
#   X34(7),Y34(7),Z34(7),X35(7),Y35(7),Z35(7),X36(7),Y36(7),Z36(7),
#   X37(7),Y37(7),Z37(7),X38(7),Y38(7),Z38(7),X39(7),Y39(7),Z39(7),
#   X40(7),Y40(7),Z40(7),X41(7),Y41(7),Z41(7),X42(7),Y42(7),Z42(7),
#   X43(7),Y43(7),Z43(7),X44(7),Y44(7),Z44(7),X45(7),Y45(7),Z45(7),
#   X46(7),Y46(7),Z46(7),X47(7),Y47(7),Z47(7),X48(7),Y48(7),Z48(7),
#   X49(7),Y49(7),Z49(7),X50(7),Y50(7),Z50(7),X51(7),Y51(7),Z51(7),
#   X52(7),Y52(7),Z52(7),X53(7),Y53(7),Z53(7)
      INTEGER REGION
      FINERX (GAMMA,RLAMB,FF)=EE*COS (3.0*GAMMA)+RR*COS (GAMMA)+
#   FF*COS (GAMMA+RLAMB)
      FINERY (GAMMA,RLAMB,FF)=EE*SIN (3.0*GAMMA)+RR*SIN (GAMMA)+
#   FF*SIN (GAMMA+RLAMB)
      XXX (Z,BETA)=Z*COS (BETA)
      YYY (ZZ,BETA)=ZZ*SIN (BETA)
```

```
BB=BBB
TRANS=ABS (TRANS)
R=RR-EE+D+AA-TRANS

IF (REGION.EQ.3) THEN

    PHI1=PI-PHIONE
    PHI2=PI-PHITWO
    YONE=ABS (R*SIN (PHI1))
    YTWO=ABS (R*SIN (PHI2))

    WIDTH=YONE+YTWO
    TCKSP=(-DEPTH/4.0-RSP)/2.0

    YONE=YONE+TRANS
    YTWO=TRANS-YTWO
    YCNTR=(YONE+YTWO)/2.0

ELSE IF (REGION.EQ.2) THEN

    WIDTH=ABS (YTWO-YONE)
    TCKSP=(-DEPTH/4.0-RSP)/2.0

    YCNTR=(YONE+YTWO)/2.0

ENDIF

DO 15 JJ=1,IEND

IF (JJ.LT.6) THEN

    X1(JJ)=-R
    Y1(JJ)=YCNTR-RSP

    X2(JJ)=-R
    Y2(JJ)=YCNTR-RSP*COS (PI/4.0)

    X3(JJ)=-R
    Y3(JJ)=YCNTR

    X4(JJ)=-R
    Y4(JJ)=YCNTR+RSP*COS (PI/4.0)

    X5(JJ)=-R
    Y5(JJ)=YCNTR+RSP

    X6(JJ)=-R
    Y6(JJ)=YCNTR+RSP*COS (PI/4.0)

    X7(JJ)=-R
    Y7(JJ)=YCNTR

    X8(JJ)=-R
    Y8(JJ)=YCNTR-RSP*COS (PI/4.0)

    X9(JJ)=-R
    Y9(JJ)=Y8(JJ)-TCKSP*COS (PI/4.0)

    X10(JJ)=-R
    Y10(JJ)=Y1(JJ)-TCKSP

    X11(JJ)=-R
    Y11(JJ)=Y2(JJ)-TCKSP*COS (PI/4.0)
```

GEN87690  
GEN87700  
GEN87710  
GEN87720  
GEN87730  
GEN87740  
GEN87750  
GEN87760  
GEN87770  
GEN87780  
GEN87790  
GEN87800  
GEN87810  
GEN87820  
GEN87830  
GEN87840  
GEN87850  
GEN87860  
GEN87870  
GEN87880  
GEN87890  
GEN87900  
GEN87910  
GEN87920  
GEN87930  
GEN87940  
GEN87950  
GEN87960  
GEN87970  
GEN87980  
GEN87990  
GEN88000  
GEN88010  
GEN88020  
GEN88030  
GEN88040  
GEN88050  
GEN88060  
GEN88070  
GEN88080  
GEN88090  
GEN88100  
GEN88110  
GEN88120  
GEN88130  
GEN88140  
GEN88150  
GEN88160  
GEN88170  
GEN88180  
GEN88190  
GEN88200  
GEN88210  
GEN88220  
GEN88230  
GEN88240  
GEN88250  
GEN88260  
GEN88270  
GEN88280  
GEN88290  
GEN88300  
GEN88310  
GEN88320

```

      X12(JJ)=-R
      Y12(JJ)=Y3(JJ)

      X13(JJ)=-R
      Y13(JJ)=Y4(JJ)+TCKSP*COS(PI/4.0)

      IF (REGION.EQ.2) THEN

        X14(JJ)=-R
        Y14(JJ)=Y5(JJ)+TCKSP

      ENDIF

      X15(JJ)=-R
      Y15(JJ)=Y6(JJ)+TCKSP*COS(PI/4.0)

      X16(JJ)=-R
      Y16(JJ)=Y7(JJ)

      X17(JJ)=-R
      Y17(JJ)=YCNTR-WIDTH/2.0

      X18(JJ)=-R
      Y18(JJ)=Y17(JJ)

      X19(JJ)=-R
      Y19(JJ)=Y17(JJ)

      X20(JJ)=-R
      Y20(JJ)=Y12(JJ)

      IF (REGION.EQ.3) THEN

        X21(JJ)=XXX(R,PHONE)
        Y21(JJ)=YYY(R,PHONE)+TRANS

        X22(JJ)=XXX(R,PHONE)
        Y22(JJ)=YYY(R,PHONE)+TRANS

        X23(JJ)=XXX(R,PHONE)
        Y23(JJ)=YYY(R,PHONE)+TRANS

        X14(JJ)=-R
        Y14(JJ)=(Y5(JJ)+Y22(JJ))/2.0

      ELSE IF (REGION.EQ.2) THEN

        X21(JJ)=-R
        Y21(JJ)=YCNTR+WIDTH/2.0

        X22(JJ)=-R
        Y22(JJ)=Y21(JJ)

        X23(JJ)=-R
        Y23(JJ)=Y21(JJ)

      ENDIF

      X24(JJ)=-R
      Y24(JJ)=Y16(JJ)

      ELSE IF (JJ.GT.5) THEN
```

GEN88330  
GEN88340  
GEN88350  
GEN88360  
GEN88370  
GEN88380  
GEN88390  
GEN88400  
GEN88410  
GEN88420  
GEN88430  
GEN88440  
GEN88450  
GEN88460  
GEN88470  
GEN88480  
GEN88490  
GEN88500  
GEN88510  
GEN88520  
GEN88530  
GEN88540  
GEN88550  
GEN88560  
GEN88570  
GEN88580  
GEN88590  
GEN88600  
GEN88610  
GEN88620  
GEN88630  
GEN88640  
GEN88650  
GEN88660  
GEN88670  
GEN88680  
GEN88690  
GEN88700  
GEN88710  
GEN88720  
GEN88730  
GEN88740  
GEN88750  
GEN88760  
GEN88770  
GEN88780  
GEN88790  
GEN88800  
GEN88810  
GEN88820  
GEN88830  
GEN88840  
GEN88850  
GEN88860  
GEN88870  
GEN88880  
GEN88890  
GEN88900  
GEN88910  
GEN88920  
GEN88930  
GEN88940  
GEN88950  
GEN88960

CALL YALPSL (EE,RR,Y1(JJ-1),ICLK,ALPHA1)	GEN88970
CALL THTASL (EE,RR,PI,ALPHA1,THETA1)	GEN88980
CALL YALPSL (EE,RR,Y2(JJ-1),ICLK,ALPHA2)	GEN88990
CALL THTASL (EE,RR,PI,ALPHA2,THETA2)	GEN89000
CALL YALPSL (EE,RR,Y3(JJ-1),ICLK,ALPHA3)	GEN89010
CALL THTASL (EE,RR,PI,ALPHA3,THETA3)	GEN89020
CALL YALPSL (EE,RR,Y4(JJ-1),ICLK,ALPHA4)	GEN89030
CALL THTASL (EE,RR,PI,ALPHA4,THETA4)	GEN89040
	GEN89050
CALL YALPSL (EE,RR,Y5(JJ-1),ICLK,ALPHA5)	GEN89060
CALL THTASL (EE,RR,PI,ALPHA5,THETA5)	GEN89070
CALL YALPSL (EE,RR,Y6(JJ-1),ICLK,ALPHA6)	GEN89080
CALL THTASL (EE,RR,PI,ALPHA6,THETA6)	GEN89090
CALL YALPSL (EE,RR,Y8(JJ-1),ICLK,ALPHA8)	GEN89100
CALL THTASL (EE,RR,PI,ALPHA8,THETA8)	GEN89110
CALL YALPSL (EE,RR,Y9(JJ-1),ICLK,ALPHA9)	GEN89120
CALL THTASL (EE,RR,PI,ALPHA9,THETA9)	GEN89130
	GEN89140
CALL YALPSL (EE,RR,Y10(JJ-1),ICLK,ALPH10)	GEN89150
CALL THTASL (EE,RR,PI,ALPH10,THET10)	GEN89160
CALL YALPSL (EE,RR,Y13(JJ-1),ICLK,ALPH13)	GEN89170
CALL THTASL (EE,RR,PI,ALPH13,THET13)	GEN89180
CALL YALPSL (EE,RR,Y14(JJ-1),ICLK,ALPH14)	GEN89190
CALL THTASL (EE,RR,PI,ALPH14,THET14)	GEN89200
CALL YALPSL (EE,RR,Y17(JJ-1),ICLK,ALPH17)	GEN89210
CALL THTASL (EE,RR,PI,ALPH17,THET17)	GEN89220
CALL YALPSL (EE,RR,Y21(JJ-1),ICLK,ALPH21)	GEN89230
CALL THTASL (EE,RR,PI,ALPH21,THET21)	GEN89240
	GEN89250
X1(JJ)=-FINERX(ALPHA1,THETA1,BBB)	GEN89260
Y1(JJ)=Y1(JJ-1)	GEN89270
	GEN89280
X2(JJ)=-FINERX(ALPHA2,THETA2,BBB)	GEN89290
Y2(JJ)=Y2(JJ-1)	GEN89300
	GEN89310
X3(JJ)=-FINERX(ALPHA3,THETA3,BBB)	GEN89320
Y3(JJ)=Y3(JJ-1)	GEN89330
	GEN89340
X4(JJ)=-FINERX(ALPHA4,THETA4,BBB)	GEN89350
Y4(JJ)=Y4(JJ-1)	GEN89360
	GEN89370
X5(JJ)=-FINERX(ALPHA5,THETA5,BBB)	GEN89380
Y5(JJ)=Y5(JJ-1)	GEN89390
	GEN89400
X6(JJ)=-FINERX(ALPHA6,THETA6,BBB)	GEN89410
Y6(JJ)=Y6(JJ-1)	GEN89420
	GEN89430
X7(JJ)=-FINERX(ALPHA3,THETA3,BBB)	GEN89440
Y7(JJ)=Y7(JJ-1)	GEN89450
	GEN89460
X8(JJ)=-FINERX(ALPHA8,THETA8,BBB)	GEN89470
Y8(JJ)=Y8(JJ-1)	GEN89480
	GEN89490
X9(JJ)=-FINERX(ALPHA9,THETA9,BBB)	GEN89500
Y9(JJ)=Y9(JJ-1)	GEN89510
	GEN89520
X10(JJ)=-FINERX(ALPH10,THET10,BBB)	GEN89530
Y10(JJ)=Y10(JJ-1)	GEN89540
	GEN89550
X11(JJ)=-FINERX(ALPHA9,THETA9,BBB)	GEN89560
Y11(JJ)=Y9(JJ)	GEN89570
	GEN89580
X12(JJ)=-FINERX(ALPHA3,THETA3,BBB)	GEN89590
Y12(JJ)=Y3(JJ)	GEN89600

```

X13(JJ)=-FINERX(ALPH13,THET13,BBB)
Y13(JJ)=Y13(JJ-1)

X14(JJ)=-FINERX(ALPH14,THET14,BBB)
Y14(JJ)=Y14(JJ-1)

X15(JJ)=-FINERX(ALPH13,THET13,BBB)
Y15(JJ)=Y13(JJ)

X16(JJ)=-FINERX(ALPHA3,THETA3,BBB)
Y16(JJ)=Y7(JJ)

X17(JJ)=-FINERX(ALPH17,THET17,BBB)
Y17(JJ)=Y17(JJ-1)

X18(JJ)=-FINERX(ALPH17,THET17,BBB)
Y18(JJ)=Y17(JJ)

X19(JJ)=-FINERX(ALPH17,THET17,BBB)
Y19(JJ)=Y17(JJ)

X20(JJ)=-FINERX(ALPHA3,THETA3,BBB)
Y20(JJ)=Y3(JJ)

IF (JJ.EQ.6) DD=BB
IF (JJ.EQ.7) DD=CC-BB

IF (REGION.EQ.3) THEN

  X21(JJ)=-FINERX(ALPH21,THET21,BBB)
  Y21(JJ)=Y21(JJ-1)+DD*SIN(PI-PHIONE)

ELSE IF (REGION.EQ.2) THEN

  X21(JJ)=-FINERX(ALPH21,THET21,BBB)
  Y21(JJ)=Y21(JJ-1)

ENDIF

X22(JJ)=-FINERX(ALPH21,THET21,BBB)
Y22(JJ)=Y21(JJ)

X23(JJ)=-FINERX(ALPH21,THET21,BBB)
Y23(JJ)=Y21(JJ)

X24(JJ)=-FINERX(ALPHA3,THETA3,BBB)
Y24(JJ)=Y3(JJ)

IF (JJ.EQ.6) BBB=CC
IF (JJ.EQ.7) BBB=BB

ENDIF

X25(JJ)=X19(JJ)
Y25(JJ)=Y19(JJ)
X26(JJ)=X20(JJ)
Y26(JJ)=Y20(JJ)
X27(JJ)=X21(JJ)
Y27(JJ)=Y21(JJ)
X28(JJ)=X23(JJ)
Y28(JJ)=Y23(JJ)
X29(JJ)=X24(JJ)
Y29(JJ)=Y24(JJ)
```

GEN89610  
GEN89620  
GEN89630  
GEN89640  
GEN89650  
GEN89660  
GEN89670  
GEN89680  
GEN89690  
GEN89700  
GEN89710  
GEN89720  
GEN89730  
GEN89740  
GEN89750  
GEN89760  
GEN89770  
GEN89780  
GEN89790  
GEN89800  
GEN89810  
GEN89820  
GEN89830  
GEN89840  
GEN89850  
GEN89860  
GEN89870  
GEN89880  
GEN89890  
GEN89900  
GEN89910  
GEN89920  
GEN89930  
GEN89940  
GEN89950  
GEN89960  
GEN89970  
GEN89980  
GEN89990  
GEN90000  
GEN90010  
GEN90020  
GEN90030  
GEN90040  
GEN90050  
GEN90060  
GEN90070  
GEN90080  
GEN90090  
GEN90100  
GEN90110  
GEN90120  
GEN90130  
GEN90140  
GEN90150  
GEN90160  
GEN90170  
GEN90180  
GEN90190  
GEN90200  
GEN90210  
GEN90220  
GEN90230  
GEN90240

```
X30(JJ)=X17(JJ)
Y30(JJ)=Y17(JJ)

Z1(JJ)=DEPTH/2.0
Z2(JJ)=DEPTH/2.0-RSP*SIN(PI/4.0)
Z3(JJ)=DEPTH/2.0-RSP
Z4(JJ)=DEPTH/2.0-RSP*SIN(PI/4.0)
Z5(JJ)=DEPTH/2.0
Z6(JJ)=DEPTH/2.0+RSP*SIN(PI/4.0)
Z7(JJ)=DEPTH/2.0+RSP
Z8(JJ)=DEPTH/2.0+RSP*SIN(PI/4.0)
Z9(JJ)=Z8(JJ)+TCKSP*SIN(PI/4.0)
Z10(JJ)=Z1(JJ)
Z11(JJ)=Z2(JJ)-TCKSP*SIN(PI/4.0)
Z12(JJ)=Z3(JJ)-TCKSP
Z13(JJ)=Z4(JJ)-TCKSP*SIN(PI/4.0)
Z14(JJ)=Z5(JJ)
Z15(JJ)=Z6(JJ)+TCKSP*SIN(PI/4.0)
Z16(JJ)=Z7(JJ)+TCKSP
Z17(JJ)=DEPTH/4.0
Z18(JJ)=DEPTH/2.0
Z19(JJ)=3.0*DEPTH/4.0
Z20(JJ)=3.0*DEPTH/4.0
Z21(JJ)=3.0*DEPTH/4.0
Z22(JJ)=DEPTH/2.0
Z23(JJ)=DEPTH/4.0
Z24(JJ)=DEPTH/4.0
Z25(JJ)=DEPTH
Z26(JJ)=DEPTH
Z27(JJ)=DEPTH
Z28(JJ)=0.0
Z29(JJ)=0.0
Z30(JJ)=0.0

IF(JJ.EQ.1) R=R-AA
IF(JJ.EQ.2) R=R-D/3.0
IF(JJ.EQ.3) R=R-D/3.0
IF(JJ.EQ.4) R=R-D/3.0
IF(JJ.EQ.5) R=R-BB
IF(JJ.EQ.6) R=R-CC

15    CONTINUE

RETURN
END

SUBROUTINE NEWFIL (STATUS,NO)

C *****
C *
C *   THIS SUBROUTINE CHECKS TO SEE IF THE FILE NAMED
C *   "GENERATE DATA A" EXISTS ON THE USER'S DISK.  IF IT
C *   DOES NOT EXIST, IT IS CREATED AND THE USER IS
C *   PROMPTED FOR THE INPUTS.
C *
C *   IF IT DOES EXIST, THEN THE USER IS
C *   NOT PROMPTED FOR THE INPUTS.  THE INPUTS ARE READ
C *   FROM THE EXISTING DATA FILE.
C *
C *****

CHARACTER FILE*20,STRING*38
INTEGER STATUS,CMSCMD
```

```
GEN90250
GEN90260
GEN90270
GEN90280
GEN90290
GEN90300
GEN90310
GEN90320
GEN90330
GEN90340
GEN90350
GEN90360
GEN90370
GEN90380
GEN90390
GEN90400
GEN90410
GEN90420
GEN90430
GEN90440
GEN90450
GEN90460
GEN90470
GEN90480
GEN90490
GEN90500
GEN90510
GEN90520
GEN90530
GEN90540
GEN90550
GEN90560
GEN90570
GEN90580
GEN90590
GEN90600
GEN90610
GEN90620
GEN90630
GEN90640
GEN90650
GEN90660
GEN90670
GEN90680
GEN90690
GEN90700
GEN90710
GEN90720
GEN90730
GEN90740
GEN90750
GEN90760
GEN90770
GEN90780
GEN90790
GEN90800
GEN90810
GEN90820
GEN90830
GEN90840
GEN90850
GEN90860
GEN90870
GEN90880
```



```
C SET STRING TO BLANKS
      CLOSE(5)
      STRING = '
      FILE='GENERATE DATA A'

C CHECK STATUS OF INPUT FILENAME
      STRING='STATE '//FILE
      STATUS=CMSCMD (STRING)

      IF (STATUS.EQ.28) THEN
        WRITE(NO,*) ' '
        WRITE(NO,*) ' THE "GENERATE DATA" FILE WAS NOT FOUND ON DISK.'
        WRITE(NO,*) ' THEREFORE, YOU WILL BE PROMPTED FOR THE INPUTS.'
        WRITE(NO,*) ' A "GENERATE DATA" FILE WILL BE CREATED FROM YOUR'
        WRITE(NO,*) ' RESPONSES SO THAT CHANGES IN THE FILE CAN EASILY'
        WRITE(NO,*) ' BE MADE AT A LATER TIME.'
        WRITE(NO,*) ' '
        GO TO 40
      ELSE IF (STATUS.EQ.36) THEN
        WRITE(NO,*) ' DISK NOT ACCESSED'
        WRITE(NO,*) ' RETURN CODE = 36'
        GO TO 50
      ELSE IF (STATUS.NE.0) THEN
15      WRITE(NO,*) ' FILE STATUS ERROR'
        WRITE(NO,*) ' RETURN CODE = ',STATUS
        GO TO 50
      ENDIF

      IF (STATUS.EQ.0) GO TO 50

C IF FILE DOES NOT EXIST, THEN CREATE IT ON UNIT #8
C
      IF (STATUS.EQ.0) GO TO 50

40      STRING='FILEDEF 8 DISK '//FILE
      STATUS=CMSCMD (STRING)
      IF (STATUS.NE.0) THEN
        WRITE(NO,*) ' FILEDEF ERROR - DEFINITION OF FILE AS UNIT 8 FAILED'
        WRITE(NO,*) ' RETURN CODE = ',STATUS
      ENDIF

      STATUS=1

50      RETURN
      END

      FUNCTION CMS CMD (COMAND)

C      INTEGER      MAX CMD
      PARAMETER      (MAX CMD = 20)
      INTEGER      MAX DBL
      PARAMETER      (MAX DBL = MAX CMD+1)

C      INTEGER      $END IT
      REAL*8      BUFFER      (MAX DBL)
      CHARACTER*8  CMD STR    (MAX CMD)
      INTEGER      CMD SUB
      INTEGER      CMS CMD
      CHARACTER*(*) COMAND
      INTEGER      FIRST
```

GEN90890  
GEN90900  
GEN90910  
GEN90920  
GEN90930  
GEN90940  
GEN90950  
GEN90960  
GEN90970  
GEN90980  
GEN90990  
GEN91000  
GEN91010  
GEN91020  
GEN91030  
GEN91040  
GEN91050  
GEN91060  
GEN91070  
GEN91080  
GEN91090  
GEN91100  
GEN91110  
GEN91120  
GEN91130  
GEN91140  
GEN91150  
GEN91160  
GEN91170  
GEN91180  
GEN91190  
GEN91200  
GEN91210  
GEN91220  
GEN91230  
GEN91240  
GEN91250  
GEN91260  
GEN91270  
GEN91280  
GEN91290  
GEN91300  
GEN91310  
GEN91320  
GEN91330  
GEN91340  
GEN91350  
GEN91360  
GEN91370  
GEN91380  
GEN91390  
GEN91400  
GEN91410  
GEN91420  
GEN91430  
GEN91440  
GEN91450  
GEN91460  
GEN91470  
GEN91480  
GEN91490  
GEN91500  
GEN91510  
GEN91520

```
      INTRINSIC      INDEX      GEN91530
      INTRINSIC      LEN        GEN91540
      INTEGER        LONG      GEN91550
      INTEGER        LST       GEN91560
      INTEGER        LST 1     GEN91570
      INTEGER        LST 2     GEN91580
      INTEGER        LST 3     GEN91590
      INTRINSIC      MIN       GEN91600
C                                     GEN91610
C      get the length of the command and initialize the start      GEN91620
      LONG = LEN(COMAND)      GEN91630
C      initialize the starting position and number of parameters  GEN91640
      FIRST = 1               GEN91650
      CMD SUB = 0             GEN91660
C      find the next blank in the command                          GEN91670
10  LST 1 = INDEX(COMAND(FIRST:),' ')      GEN91680
      IF (LST 1 .EQ. 1)      GEN91690
      > THEN                  GEN91700
C          first column was blank, move to next column            GEN91710
          FIRST = FIRST + 1      GEN91720
          GO TO 100              GEN91730
      ENDIF                  GEN91740
C      look for a left or right paren                              GEN91750
      LST 2 = INDEX(COMAND(FIRST:),'(')    GEN91760
      LST 3 = INDEX(COMAND(FIRST:),')')    GEN91770
      LST = LONG-FIRST+2          GEN91780
C      set LST to the location of the first "(", ")", or " "      GEN91790
      IF (LST 1 .GT. 0) LST = MIN(LST,LST 1) GEN91800
      IF (LST 2 .GT. 0) LST = MIN(LST,LST 2) GEN91810
      IF (LST 3 .GT. 0) LST = MIN(LST,LST 3) GEN91820
      LST = LST + FIRST - 2      GEN91830
C      pick up next parameter, if it is there                     GEN91840
      IF (LST .GE. FIRST)      GEN91850
      > THEN                  GEN91860
          CMD SUB = CMD SUB + 1      GEN91870
C          make sure we have room for this parameter              GEN91880
          IF (CMD SUB .GT. MAX CMD) GO TO 200 GEN91890
          CMD STR(CMD SUB)=COMAND(FIRST:LST) GEN91900
      ENDIF                  GEN91910
C      skip over trailing blank, if any                            GEN91920
      IF (LST+1 .LE. LONG)      GEN91930
      > THEN                  GEN91940
          IF (COMAND(LST+1:LST+1) .NE. ' ') GEN91950
      > THEN                  GEN91960
          CMD SUB = CMD SUB + 1      GEN91970
          IF (CMD SUB .GT. MAX CMD) GO TO 200 GEN91980
          CMD STR(CMD SUB)=COMAND(LST+1:LST+1) GEN91990
      ENDIF                  GEN92000
      ENDIF                  GEN92010
      FIRST = LST+2            GEN92020
C      loop back if there is more to process                       GEN92030
100 IF (FIRST .LE. LONG) GO TO 10      GEN92040
C      no more to process, call $ENDIT to set it up for CM$CMD    GEN92050
      CMSCMD = $ENDIT (CMD STR,BUFFER,CMD SUB) GEN92060
      RETURN                  GEN92070
200 CMSCMD = -1               GEN92080
      RETURN                  GEN92090
      END                    GEN92100
      FUNCTION $ENDIT (CMD STR,BUFFER,BUF SIZ) GEN92110
C                                     GEN92120
C      this function will make sure that the command is properly  GEN92130
C      aligned for the SVC in CM$CMD                               GEN92140
C                                     GEN92150
      INTEGER          $END IT      GEN92160
```

```

      INTEGER      BUF SIZ                      GEN92170
      REAL*8       BUFFER      (BUF SIZ)        GEN92180
      INTEGER      CM$CMD                      GEN92190
      LOGICAL*1    CMD STR      (8,BUF SIZ)     GEN92200
      REAL*8       D TEMP                      GEN92210
      REAL*8       EOM FLG                      GEN92220
      INTEGER      I                          GEN92230
      INTEGER      J                          GEN92240
      LOGICAL*1    L TEMP      (8)             GEN92250
C
      DATA EOM FLG /ZFFFFFFFFFFFFFFFFF/        GEN92260
C
      EQUIVALENCE (L TEMP,D TEMP)              GEN92270
C
      DO 20 I = 1,BUF SIZ                      GEN92280
        DO 10 J = 1,8                          GEN92290
          L TEMP(J)=CMD STR(J,I)              GEN92300
        CONTINUE                              GEN92310
      CONTINUE                                GEN92320
      BUFFER(I) = D TEMP                      GEN92330
      CONTINUE                                GEN92340
      BUFFER(BUF SIZ+1) = EOM FLG             GEN92350
      $END IT = CM$CMD(BUFFER)                GEN92360
      RETURN                                  GEN92370
      END                                     GEN92380
C
      *****                                GEN92390
C      *                                GEN92400
C      *                                GEN92410
C      *      THIS SUBROUTINE WRITES A PROGRAM FILE THAT *      GEN92420
C      *      CONTAINS THE COMMANDS THAT WILL MERGE THE *      GEN92430
C      *      COINCIDENT NODES THAT EXIST IN THE MODEL. *      GEN92440
C      *      THE TOLERAENCE IS SET AT 0.01 INCHES. *      GEN92450
C      *                                *      GEN92460
C      *                                *      GEN92470
C      *****                                *      GEN92480
C
      SUBROUTINE COIN (NO)                      GEN92490
      CHARACTER *10 NOAL,SLASH,GENERL,YES      GEN92500
      NOAL='4 -1NOAL'                          GEN92510
      SLASH='1 -1/'                            GEN92520
      GENERL='10 -1'                          GEN92530
      YES='3 -1YES'                            GEN92540
      WRITE(15,30) SLASH                      GEN92550
      FORMAT(2X,A5)                          GEN92560
      WRITE(15,35) GENERL                      GEN92570
      FORMAT(1X,A5,'T')                      GEN92580
      WRITE(15,40) GENERL                      GEN92590
      FORMAT(1X,A5,'MC')                     GEN92600
      WRITE(15,45) GENERL                      GEN92610
      FORMAT(1X,A5,'CN')                     GEN92620
      WRITE(15,50) GENERL                      GEN92630
      FORMAT(1X,A5,'ALL')                     GEN92640
      WRITE(15,55) GENERL                      GEN92650
      FORMAT(1X,A5,'0.01')                     GEN92660
      WRITE(15,60) YES                        GEN92670
      FORMAT(2X,A8)                          GEN92680
      GEN92690
      GEN92700
      GEN92710
      GEN92720
      GEN92730
      GEN92740
      GEN92750
      GEN92760
      GEN92770
      GEN92780
      GEN92790
      GEN92800
```

WRITE(15,60) YES	GEN92810
WRITE(15,60) YES	GEN92820
WRITE(15,60) YES	GEN92830
WRITE(15,60) YES	GEN92840
WRITE(15,60) YES	GEN92850
WRITE(15,60) YES	GEN92860
WRITE(15,60) YES	GEN92870
WRITE(15,60) YES	GEN92880
WRITE(15,60) YES	GEN92890
WRITE(15,60) YES	GEN92900
WRITE(15,60) YES	GEN92910
WRITE(15,60) YES	GEN92920
WRITE(15,60) YES	GEN92930
WRITE(15,60) YES	GEN92940
WRITE(15,60) YES	GEN92950
WRITE(15,60) YES	GEN92960
WRITE(15,60) YES	GEN92970
WRITE(15,60) YES	GEN92980
WRITE(15,60) YES	GEN92990
WRITE(15,60) YES	GEN93000
WRITE(15,60) YES	GEN93010
WRITE(15,60) YES	GEN93020
WRITE(15,60) YES	GEN93030
WRITE(15,60) YES	GEN93040
WRITE(15,60) YES	GEN93050
WRITE(15,60) YES	GEN93060
WRITE(15,60) YES	GEN93070
WRITE(15,60) YES	GEN93080
WRITE(15,60) YES	GEN93090
WRITE(15,60) YES	GEN93100
WRITE(15,60) YES	GEN93110
WRITE(15,60) YES	GEN93120
WRITE(15,60) YES	GEN93130
WRITE(15,60) YES	GEN93140
WRITE(15,60) YES	GEN93150
WRITE(15,60) YES	GEN93160
WRITE(15,60) YES	GEN93170
WRITE(15,60) YES	GEN93180
WRITE(15,60) YES	GEN93190
WRITE(15,60) YES	GEN93200
WRITE(15,60) YES	GEN93210
WRITE(15,60) YES	GEN93220
WRITE(15,60) YES	GEN93230
	GEN93240
WRITE(15,30) SLASH	GEN93250
WRITE(15,35) GENERL	GEN93260
	GEN93270
WRITE(15,65) GENERL	GEN93280
FORMAT(1X,A5,'MA')	GEN93290
	GEN93300
WRITE(15,70) GENERL	GEN93310
FORMAT(1X,A5,'NO')	GEN93320
	GEN93330
WRITE(15,75) GENERL	GEN93340
FORMAT(1X,A5,'DEL')	GEN93350
	GEN93360
WRITE(15,80)	GEN93370
FORMAT(2X,'4 27 K')	GEN93380
	GEN93390
WRITE(15,85) GENERL	GEN93400
FORMAT(1X,A5,'LABEL')	GEN93410
	GEN93420
WRITE(15,90) GENERL	GEN93430
FORMAT(1X,A5,'1 50000 1')	GEN93440

WRITE(15,60) YES

RETURN  
END

GEN93450  
GEN93460  
GEN93470  
GEN93480  
GEN93490